(19) Europäisches Patentamt
Europ an Patent Office
Office uropéen des br v t

(11) EP 1 162 572 A2

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
12.12.2001 Bulletin 2001/50

(51) Int Cl.[7]: **G06T 7/00**

(21) Application number: **01304956.4**

(22) Date of filing: **06.06.2001**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **06.06.2000 US 589046**
11.09.2000 US 659415

(71) Applicant: **Agilent Technologies, Inc. (a Delaware corporation)**
Palo Alto, CA 94303 (US)

(72) Inventors:
- **Yakhini, Zohar H.**
  Zikron Ya'acov 30900 (IL)
- **Enderwick, Cynthia Y.**
  San Jose, CA 95112 (US)
- **Delenstarr, Glenda C.**
  Belmont, CA 94002 (US)
- **Wolber, Paul K.**
  Los Altos, CA 94022 (US)
- **Sampas, Nicholas M.**
  San Jose, CA 95125 (US)

(74) Representative: **Jehan, Robert et al**
Williams, Powell & Associates,
4 St Paul's Churchyard
London EC4M 8AY (GB)

(54) Method and system for evaluating orientation of molecular array

(57) A method and system for evaluating an orientation of a molecular array having features arranged in a pattern obtain an image of the molecular array by scanning the molecular array to determine data signals emanating from discrete positions on a surface of the molecular array. An actual result of a function on pixels of the image which pixels lie in a second pattern, is calculated. This actual result is compared with an expected result which would be obtained if the second pattern had a predetermined orientation on the array. Array orientation can then be evaluated based on the result.

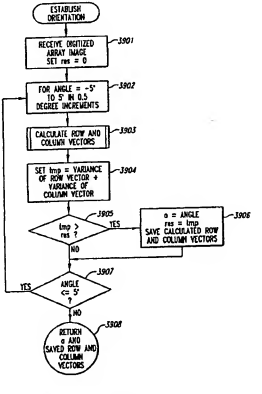


Fig. 39

EP 1 162 572 A2

Description

[0001] The present invention relates to the evaluation of the orientation of a molecular array, for example for processing a scanned image of a molecular array in order to index the regions of the image that correspond to features of the molecular array and to extract data from indexed positions within the scanned image that correspond to optical or radiometric signals emanating from features of the molecular array.

[0002] Molecular arrays are widely used and increasingly important tools for rapid hybridization analysis of sample solutions against hundreds or thousands of precisely ordered and positioned features containing different types of molecules within the molecular arrays. Molecular arrays are normally prepared by synthesizing or attaching a large number of molecular species to a chemically prepared substrate such as silicone, glass, or plastic. Each feature, or element, within the molecular array is defined to be a small, regularly-shaped region on the surface of the substrate. The features are arranged in a regular pattern. Each feature within the molecular array may contain a different molecular species, and the molecular species within a given feature may differ from the molecular species within the remaining features of the molecular array. In one type of hybridization experiment, a sample solution containing radioactively, fluorescently, or chemoluminescently labeled molecules is applied to the surface of the molecular array. Certain of the labeled molecules in the sample solution may specifically bind to, or hybridize with, one or more of the different molecular species that together comprise the molecular array. Following hybridization, the sample solution is removed by washing the surface of the molecular array with a buffer solution, and the molecular array is then analyzed by radiometric or optical methods to determine to which specific features of the molecular array the labeled molecules are bound. Thus, in a single experiment, a solution of labeled molecules can be screened for binding to hundreds or thousands of different molecular species that together comprise the molecular array. Molecular arrays commonly contain oligonucleotides or complementary deoxyribonucleic acid ("cDNA") molecules to which labeled deoxyribonucleic acid ("DNA") and ribonucleic acid ("RNA") molecules bind via sequence-specific hybridization.

[0003] Generally, radiometric or optical analysis of the molecular array produces a scanned image consisting of a two-dimensional matrix, or grid, of pixels, each pixel having one or more intensity values corresponding to one or more signals. Scanned images are commonly produced electronically by optical or radiometric scanners and the resulting two-dimensional matrix of pixels is stored in computer memory or on a non-volatile storage device. Alternatively, analog methods of analysis, such as photography, can be used to produce continuous images of a molecular array that can be then digitized by a scanning device and stored in computer memory or in a computer storage device.

[0004] Figure 1 shows a generalized representation of a molecular array. Disk-shaped features of the molecular array, such as feature 101, are arranged on the surface of the molecular array in rows and columns that together comprise a two-dimensional matrix, or grid. Features in alternative types of molecular arrays may be arranged to cover the surface of the molecular array at higher densities, as, for example, by offsetting the features in adjacent rows to produce a more closely packed arrangement of features. Radiometric or optical analysis of a molecular array, following a hybridization experiment, results in a two-dimensional matrix, or grid, of pixels. Figure 2 illustrates the two-dimensional grid of pixels in a square area of a scanned image encompassing feature 101 of Figure 1. In Figure 2, pixels have intensity values ranging from 0 to 9. Intensity values of all non-zero pixels are shown in Figure 2 as single digits within the pixel. The non-zero pixels of this scanned image representing feature 101 of Figure 1 inhabit a roughly disk-shaped region corresponding to the shape of the feature. The pixels in a region surrounding a feature generally have low or 0 intensity values due to an absence of bound signal-producing radioactive, fluorescent, or chemoluminescent label molecules. However, background signals, such as the background signal represented by non-zero pixel 202, may arise from non-specific binding of labeled molecules due to imprecision in preparation of molecular arrays and/or imprecision in the hybridization and washing of molecular arrays, and may also arise from imprecision in optical or radiometric scanning and various other sources of error that may depend on the type of analysis used to produce the scanned image. Additional background signal may be attributed to contaminants in the surface of the molecular array or in the sample solutions to which the molecular array is exposed. In addition, pixels within the disk-shaped image of a feature, such as pixel 204, may have 0 values or may have intensity values outside the range of expected intensity values for a feature. Thus, scanned images of molecular array features may often show noise and variation and may depart significantly from the idealized scanned image shown in Figure 1.

[0005] Figure 3 illustrates indexing of a scanned image produced from a molecular array. A set of imaginary horizontal and vertical grid lines, such as horizontal grid line 301, are arranged so that the intersections of vertical and horizontal grid lines correspond with the centers of features. The imaginary grid lines establishes a two-dimensional index grid for indexing the features. Thus, for example, feature 302 can be specified by the indices (0,0). For alternative arrangements of features, such as the more closely packed arrangements mentioned above, a slightly more complicated indexing system may be used. For example, feature locations in odd-indexed rows having a particular column index may be understood to be physically offset horizontally from feature locations having the same column index in even-indexed rows. Such horizontal offsets occur, for example, in hexagonal, closest-packed arrays of features.

[0006] In order to interpret the scanned image resulting from optical or radiometric analysis of a molecular array, the

scanned image needs to be processed to: (1) index the positions of features within the scanned image; (2) extract data from th featur s and determin th magnitud s of background signals; (3) comput , for each signal, background subtracted magnitud s for ach feature; (4) normalize signals produc d from diff rent types of analysis, as, for xample, dye normalization of optical scans conduct d at different light wavelengths to normalize diff rent response curves produced by chromophores at different wavelengths; and (5) det rmine th ratios of background-subtract d and normalized signals for each feature while also determining a statistical measure of the variability of the ratios or confidence intervals related to the distribution of the signal ratios about a mean signal ratio value. These various steps in the processing of scanned images produced as a result of optical or radiometric analysis of molecular arrays together comprise an overall process called feature extraction.

[0007] The present invention seeks to provide improved analysis of a molecular array.

[0008] According to an aspect of the present invention, there is provided a method of automated extraction of data as specified in claim 1.

[0009] According to another aspect of the present invention, there is provided a system for automated extraction of data as specified in claim 14.

[0010] The preferred embodiment can provide automated feature extraction which can produce enormous savings in the time and cost of using molecular arrays for chemical and biological analysis. It can also eliminate inconsistencies caused by user error and can greatly increase the reproducibility and objectivity of feature extraction.

[0011] In the preferred embodiment there is provided a method of evaluating an actual orientation of a molecular array having features which may be arranged in a pattern (for example, on a rectilinear grid). In this method, one or more images of the molecular array are obtained. Such images may be produced by scanning the molecular array to determine data signals emanating from discrete positions on a surface of the molecular array, or by other means. An actual result of a function is calculated on positions (for example, pixels) of an image, which positions lie in a second pattern (for example, along one or more paths such as along the expected positions of array features). This actual result is compared with an expected result which would be obtained if the second pattern had a predetermined orientation on the array (for example, superimposed over at least part of the array). An actual array orientation may be evaluated based upon the results of the comparison. For example, if there is more than a predetermined difference (a "tolerance") between the actual and expected results, this could be taken as an indication that the array does not have the expected orientation in a scanner. Before trying to interpret the image data further, the orientation of the second pattern on the array can be altered and the comparison step repeated, and these comparison and second pattern reorientation steps repeated in further iterations as often as necessary until the actual and expected results are within the predetermined difference (at which point the actual and expected orientations should be the same, within the predetermined tolerance). The greater the points on the second pattern, the greater the accuracy of the orientation information that can generally be obtained from the comparison.

[0012] In a particular implementation where the features of the molecular array are arranged on a rectilinear grid, the second pattern may be a rectilinear grid of rows and columns which would lie on the rows and columns of the rectilinear grid of the array when the second pattern and array are superimposed. In this case then, the calculation may be a function executed along the rows and columns of the second pattern (for example, to obtain row and column vectors). The actual result may be compared to the expected result quantitatively or only qualitatively. For example, in the case of row and column vectors the comparison may be a comparison of the actual vector shapes with the expected vector shapes if the second pattern had the predetermined orientation on the array (the predetermined orientation, for example, rows and columns of the second pattern being aligned over rows and columns of features).

[0013] Typically, the orientation referenced is a rotational orientation (that is a rotational orientation about an axis normal to the array) although the orientation could be one or more positions in space (for example, sideways or up and down displacement). In any aspect of the method, the calculating and comparing steps may optionally be repeated by changing the orientation of the pattern on the image until a match (either exactly or within a predetermined tolerance) between the actual and expected results is obtained. The difference between the orientation of the second pattern on the image and the predetermined orientation on the array when the match is obtained, may be used in the evaluation step as a measure of the orientation of the array. Alternatively (or even additionally), a given actual result may be compared with different expected results based on different predetermined orientations of the second pattern on the array, until the match is obtained. Information on the orientation of the array can then be used in the extraction of data from the array.

[0014] One embodiment provides a method and system for automated feature extraction from scanned images produced by optical, radiometric, or other types of analysis of molecular arrays. First, horizontal and vertical projections of pixel values, called row and column vectors, are computationally produced from the scanned image. The row and column vectors are analyzed to determine the positions of peaks, and the positions of the first and last peaks in the row and column vectors are used to estimate the positions of the corner features within the scanned image. Typically, bright control features, i.e. features designed to hybridize to labeled sample molecules of any sample solution to which a molecular array is exposed, ar plac d on the border of the molecular array to facilitat this process. When necessary,

row and column vectors can be calculated over a range of rotations of a two-dimensional, orthogonal coordinate system in ord r to select the most favorable rotation angl at which to fix the coordinate system. Analysis of regions of the scanned image representing the corner features can be used to more xactly locate the positions of the com r features. Then, using the established positions of the corner features, an initial coordinate syst m is computationally established for the scanned image. Using the initial coordinate system, the centroids of features producing strong signals or, in other words, pixels having high signal-to-noise ratios and located close to expected positions in the scanned image, are determined, and a regression analysis is used to refine the coordinate system to best correspond to the determined positions of the strong features. The refined coordinate system is employed to locate the positions of weak features and the positions of the background regions local to each feature. Next, a process is used to analyze various different signals generated by different analytical methods in order to select the most reliable portions of each feature and the local background surrounding the feature for subsequent signal extraction and signal variability determinations. For example, the fluorescence of hybridized labeled molecules may be measured at green light wavelengths and at red light wavelengths, with the intensities produced at each position of the surface of the molecular array at red and green wavelengths corresponding to two different signals. Finally, signal data and variability of signal data are extracted from the reliable regions of each feature and each local background region of the scanned image.

[0015] It will also be appreciated throughout the present application that where steps are referenced as being implemented by a computer program, any such steps can also be implemented by hardware or hardware/software combinations which can perform the steps. Also, an "image" in relation to an array is a term which includes data on the position of the features regardless of how such data was obtained (for example, by scanning with a laser beam, or by some other means). Furthermore, wherever a function is referenced, this can for example be a summation. Likewise, wherever summation is referenced, other suitable functions might be used than a simple summation. For example, a weighted summation could be used at those locations on the second pattern which will be superimposed on reference marks or features with expected stronger signals (such as control features) relative to other features, when the second pattern and array are superimposed. When reference marks are present, it will be appreciated that they can be detected and an orientation of the array (such as rotational orientation) can be evaluated based on the detected positions of the reference marks. For example, an approximate indication of array orientation can be based on the detected positions of the reference marks followed by further evaluating a refined array orientation using the above described comparison and re-orientation procedure, such that fewer iterations of the comparison and second pattern re-orientation procedure may be required.

[0016] An embodiment of the present invention is described below, by way of example only, with reference to the accompanying drawings, in which:

Figure 1 shows a generalized representation of a molecular array.
Figure 2 illustrates a two-dimensional grid of pixels in a square area of a scanned image.
Figure 3 illustrates indexing of a scanned image produced from a molecular array.
Figure 4 illustrates an initial step in the determination of the positions of corner features of a scanned image of a molecular array.
Figure 5 illustrates a numerical calculation of a portion of the column vector corresponding to a single feature.
Figure 6A shows the initial column vector computed from a pixel grid.
Figure 6B shows the column vector of Figure 6A following smoothing.
Figure 7 graphically illustrates the initial column vector of Figure 6A.
Figure 8 graphically illustrates the smoothed column vector of Figure 6B.
Figure 9 illustrates a Gaussian filter applied to an initial column vector to produce a smoothed column vector.
Figures 10-12 illustrate the effects of rotation of a feature coordinate grid with respect to a pixel grid.
Figure 13 illustrates a computational technique for rotating a feature coordinate grid with respect to a pixel grid so that the two grids are aligned.
Figure 14 illustrates one technique for determining corner feature positions in pixel coordinates.
Figure 15 illustrates a corner feature estimation technique more suitable for densely-packed molecular arrays.
Figures 16-17 illustrate one method for threshold determination for blob analysis.
Figures 18 and 19 illustrate a second technique for choosing a threshold for blob analysis.
Figure 20 illustrates the centroid pixel of the largest blob shown in Figure 19.
Figure 21 illustrates an initial feature coordinate grid overlaying the scanned image of a molecular array.
Figure 22 illustrates a linear regression analysis used in one embodiment of the present invention.
Figure 23A shows a small 9 x 12 molecular array.
Figure 23B shows a strong matrix corresponding to the 9 x 12 molecular array of Figure 23A.
Figure 24 shows a pseudo-inverse strong matrix calculated from the strong matrix shown in Figure 23B.
Figure 25 shows a centroid matrix that includes the coordinates of the c ntroids of the strong features of the 9 x 12 matrix shown in Figure 23A.

Figure 26 shows a coefficient matrix calculated by multiplying the transpose of the pseudo-inverse matrix (Figure 24) by the centroid matrix (Figure 25).
Figures 27A-B show the fitted positions matrix calculated by multiplying the transpose of a feature indices matrix by the coefficient matrix shown in Figure 26.
Figure 28 illustrates a technique for outlier rejection.
Figure 29 shows the green signal intensity values for a feature.
Figure 30 shows the red intensity values for a feature.
Figure 31 illustrates a number of local backgrounds.
Figure 32 illustrates a minimum feature.
Figure 33 is a flow-control diagram for the routine "refine feature coordinates."
Figure 34 is a flow-control diagram of the routine "prepare binmap."
Figure 35 is a flow-control diagram for the routine "find equivalences."
Figure 36 is a flow-control diagram for the routine "set equivalence."
Figure 37 is a flow-control diagram for the routine "find blob."
Figure 38 is a flow-control diagram for the routine "set feature coordinates."
Figure 39 is the flow-control diagram of the routine "establish orientation" that determines a correct orientation for an initial indexing grid, as discussed above with reference to 4-13.
Figure 40 is a flow-control diagram for the routine "establish corners."
Figure 41 is a flow-control diagram for the routine "establish grid."

[0017] When one item is indicated as being "remote" from another, this is referenced that the two items are at least in different buildings, and may be at least one mile, ten miles, or at least one hundred miles apart. "Communicating" information references transmitting the data representing that information over a suitable communication channel (for example, a private or public network). "Forwarding" an item refers to any means of getting that item from one location to the next, whether by physically transporting that item and includes, at least in the case of data, physically transporting a medium carrying the data or communicating the data. An array "package" may be the array plus only a substrate on which the array is deposited, although the package may include other features (such as a housing with a chamber). A " set" or a " sub-set" may have one or more members (for example, one or more features). Reference to a singular item, includes the possibility that there are plural of the same items present.

[0018] The described embodiment is directed towards automated feature extraction from scanned images of molecular arrays. Automated feature extraction includes: (1) a determination of the approximate positions of the features, for example by determining the positions of corner features within the scanned image; (2) generation of an initial coordinate system for the scanned image, for example, by using the positions of corner features, or by alternative means, including using mechanically precise positioning of features on the molecular array and of the molecular on the detection device and by using fiducial reference marks incorporated in the molecular array and detected independently, but in spatial alignment with detection of chemical features, and refinement of the initial coordinate system to produce a refined coordinate system; (3) determination of reliable regions of the scanned image from which to extract signal data; (4) and extraction of signal data from the features and local background regions of the scanned image of the molecular array. Each of these four components of automated feature extraction will be first discussed in four subsections, below, using illustrated examples and mathematical expressions. Following the first four subsections, an illustrative implementation of an automated feature extraction system is provided in a fifth subsection. Although the above-listed four components are described in terms of a combined, comprehensive feature-extraction implementation, each component may also be applied separately to discrete data extraction and processing problems.

[0019] It should be noted that the term "signal" is employed in the following discussion to indicate the data collected from features of a molecular array by a particular type of analysis. For example, if molecules binding to features are labeled with chromophores, and optical scans at red and green wavelengths of light are used to extract data from the molecular array, then the data collected during the optical scan at the green wavelength may be considered to be the green signal and data collected during the optical scan at the red wavelength may be considered to be the red signal. Signals of additional colors may be collected using additional dye sets. The practical limit to the number of types of signals, or colors, that may be collected is the number of emission spectra that can be independently observed. Using existing technologies, as many as twelve emission spectra may be independently observed. By using combinations of narrow band dyes, such as quantum dots, greater than twelve emission spectra may possibly be independently observed. Another type of signal may be collected by radiometric analysis of the molecular array for localized emission of one type of radiation with minimum energy levels for detection.

[0020] The term "signal" is also used to refer to data extracted from a particular feature using a particular type of analysis. It will be clear, from the context in which the term "signal" is used, below, whether "signal" refers to a cumulative set of data collected from a molecular array via a particular type of analysis or to data collected from a particular feature of a molecular array via a particular type of analysis.

Location f th Positi ns f Corner Features

[0021] Figure 4 illustrates an initial step in th d termination of the positions of corner f atures of a scanned imag of a molecular array. The scanned image of the molecular array 402 is represented in Figure 4 as a grid of pixels, with the higher-intensity pixels corresponding to features illustrated as dark circles, such as the disk-shaped group of pixels 404 corresponding to a corner feature. The intensity values of the pixels are projected horizontally to form a row vector 406 and are projected vertically to produce a column vector 408. The row vector 406 is illustrated as a two-dimensional graph, where the total projected intensity value is plotted in the vertical direction 409 and the position of each row of pixels is plotted in the horizontal direction 410. Projection of the intensity values of the pixels along the rows produces a wave-like graph 412, in which the peaks correspond to rows of pixels that include the centers of features and the troughs correspond to rows of pixels between features. Summing the intensity values of pixels along columns analogously produces the column vector 408.

[0022] Figure 5 illustrates a numerical calculation of a portion of the column vector corresponding to a single feature. A column vector is calculated for all features, as described in the above paragraph, and contains a number of peaks. However, for the sake of simplicity of illustration, Figure 5 shows pixel intensity values for a single feature, and the method of column vector calculation will therefore produce a single peak corresponding to the single feature. Figure 5 shows a grid of pixels 502 representing a square region of a scanned image encompassing a single feature. The intensity values of all the pixels in each column of the grid of pixels 502 are summed, and the sums are entered into the linear array 504. For example, column 506 includes four non-zero pixels having intensity values of: 1,1,2, and 1. Thus, summing all of the intensity values of the pixels in column 506 produces the sum 5 (508 in Figure 5) in the third element of array 504 corresponding to column 506. Note that, in Figure 5 and in all subsequent pixel illustrations, 0 intensity values are not explicitly shown, and pixels having intensity value of "0" are shown as blank, or unfilled, squares, such as pixel 510. Note also that other operations such as averaging may be performed as an alternative to summing of rows and columns of pixels to create the row and column vectors.

[0023] Next, the row and column vectors (406 and 408 in Figure 4) may be computationally smoothed, for example, by applying a Gaussian filter. The smoothing process is illustrated in Figures 6-9 using the example of Figure 5. As with Figure 5, Figures 6-9 use a single-peak column vector calculated from a single feature for the sake of simplicity of illustration. In general, the smoothing process is applied to multi-peak row and column vectors, such as those illustrated in Figure 4. Figure 6A shows the initial column vector computed from the pixel grid of Figure 5, with the column positions shown as the left-hand column of Figure 6A 602 and the sum of the intensities of the pixels in each column shown in column 604. Thus, for example, it can be seen in Figure 6A that the sum of the intensity values of the pixels in column 14 (606 in Figure 6) is 169 (608 in Figure 6). Figure 6B shows the column vector of Figure 6A following smoothing. Figure 7 graphically illustrates the initial column vector of Figure 6A, and Figure 8 graphically illustrates the smoothed column vector of Figure 6B. In Figures 7 and 8, the vertical axes (702 and 802 in Figures 7 and 8, respectively) represent the sum of pixel intensities in a column, and the horizontal axes (704 and 804 in Figure 7 and 8, respectively) represent the position of the columns measured in units of pixels. Comparing Figure 7 to Figure 8, it can be observed that the smoothing process eliminates or smoothes the occasional sharp peaks, such as peaks 706-711 in Figure 7.

[0024] Figure 9 illustrates a Gaussian filter applied to an initial column vector to produce a smoothed column vector. The Gaussian filter is a discrete function with values for a range of pixel positions within a column or row vector. The value for the sum of pixel intensities at pixel position 0 (902 in Figure 9) is 0.5 and the value of the Gaussian filter function for pixel positions 1 and -1 (904 and 906 in Figure 9, respectively) are both 0.25. This Gaussian filter is convolved with an initial column vector or an initial row vector to produce a smooth column vector or smooth row vector, respectively. The convolution operation for discrete functions involves overlaying the central position of the Gaussian filter at each position, or value, in a row or column vector and calculating the smoothed row or column vector value for that position by multiplying the Gaussian filter function with the row or column vector, respectively. Thus, for example, the value 4.75 (610 in Figure 6B) corresponding to column position 2 (612 in Figure 6B) is generated by centering the Gaussian filter of Figure 9 over the value representing the sum of intensities for column 2 (508 in Figure 5) of the initial column vector (504 in Figure 5) and multiplying the resulting Gaussian filter function and the initial column vector. Because the Gaussian filter is 0 for all pixel locations other than -1, 0, and 1, multiplication of a Gaussian filter centered on element 508 of the column vector of Figure 5 produces the following result: 0.5 * 5 + 0.25 * 9 = 4.75. Many other smoothing filter techniques may be employed, including the median filter technique, to be discussed below.

[0025] In certain cases, there may be slight rotational discrepancies between the orientation of the rectilinear grid of pixels and the orientation of the rectilinear coordinate grid that describes and indexes the centers of features within the scanned image. Such discrepancies may arise when the rows and columns of features of the molecular array are rotated with respect to the horizontal and vertical axes of the scanning device. Figures 10-12 illustrate the effects of rotation of the feature coordinate grid with respect to the pixel grid. In Figure 10A, the pixel grid and the feature coordinate grid are both aligned, producing a column vector shown in Figure 10B with sharp and distinct peaks and troughs.

In Figure 11A, the pixel grid is rotated slightly counterclockwise with respect to the feature grid, producing a column vector, shown in Figur 11B, with r lativ ly shallow p aks and troughs, sinc r lativ ly high-int nsity pixels occur in each column of pixels. In Figure 12A, the pixel grid is rotat d slightly clockwise with r spect to th feature coordinat grid, producing a column vector, shown in Figure 12C, with shallow peaks and troughs. Thus, to correctly ori nt the feature coordinate grid with r sp ct to th pixel grid, row and column vectors can b calculated for a range of rotations of the feature grid with respect to the pixel grid, and the rotation producing the column and row vectors having the most distinct peaks and troughs can be determined to be the correct orientation of the feature coordinate grid with respect to the pixel grid. This technique need be employed only when the rotational discrepancy, for a rectilinear array, is greater than θ calculated as follows: sin θ

$$\sin\theta \approx 2S/L$$

where:

S = spacing of features in the *x* direction
L = length of the molecular array in the *y* direction

[0026] Figure 13 illustrates a computational technique for rotating the feature coordinate grid with respect to the pixel grid. Using the rotation angle determined by the method described in Figure 10-12, represented in Figure 13 by the symbol "θ," the positions within the scanned image can be determined with reference to a rotated pixel grid by multiplying the original coordinates of each point by a rotational matrix to generate new, transformed coordinates for the rotated pixel grid according to the following matrix equation:

$$(x_s, y_s)\begin{pmatrix}\cos\theta & \sin\theta \\ -\sin\theta & \cos\theta\end{pmatrix} = (x_n, y_n)$$

[0027] However, there may be areas within the new coordinate space, such as area 1302, that do not correspond to physical regions within the scanned image, and conversely, there may be regions in the original scanned image, such as region 1304, which lie outside the new, rotated pixel coordinate system. If the angle of rotation is small, the amount of nonoverlapping regions, such as regions 1302 and 1304, may be relatively insignificant, and may be computationally accounted for by simply decreasing the dimensions, in pixels, of the rotated coordinate system.

[0028] In non-rectilinear feature arrangements, similar techniques can be applied to orient an indexing scheme with respect to scanned images of the surface of a molecular array. For example, if the features are arranged in circles of increasing radii from a central point, then radial vectors may be calculated at various angles about a located central feature. In general, an orientation providing summed intensity profiles that most closely corresponds to idealized intensity profiles may be chosen for an appropriate indexing scheme.

[0029] To facilitate corner feature position estimation, the corner features of molecular arrays may be manufactured to contain a control substance that produces a strong signal, referred to as a "positive control," as well as another control substance that produces a weak signal near background level, referred to as "a negative control." The appropriate placement of positive and negative controls in the corners provide "soft fiducials". Placement information for soft fiducials is stored in a design file. Each molecular array may be associated with a design file that can be accessed electronically and that annotates the probe type and probe sequences for each feature position on the array.

[0030] An alternative method of determining corners of an array uses the design file associated with a molecular array in order to choose a sub-section of the array on which to perform corner analysis. For example, a small subsection in the upper left corner of a molecular array may be selected by querying the design file to find a region with a desired number of positive controls and a desired number of negative controls. This process may then be repeated for the lower right corner. A mask may then be made with these two sub-sections held in the correct distance from one another, in accordance with information stored in the design file. Convolutions can then be performed in all directions, including horizontal, vertical, diagonal, and rotational directions, using an algorithm such as a minimizing least squares algorithm in order to find the area of the scan which best matches this mask. This method may be faster than the algorithms discussed above, since the above discussed algorithms perform calculations on the entire array. In addition, this method may be more robust with regard to large signal spikes in the background regions and other signal artifacts on the molecular array. If an area of very high background or other array artifact were present in the initial mask, an acceptable solution may not be produced by th previously discussed algorithms. Th current algorithm detects the goodness of fit and, if necessary, iterate. For instance, increasingly large sub-sections may b iteratively chos n for proc ssing until

an acceptable solution were found.

[0031] Once the pixel grid and feature coordinate grid are aligned, the positions of the corner pixels can be estimated from the smoothed column and row vectors. Figure 14 illustrates one technique for determining corner feature positions in pixel coordinates. Assuming that the vector in Figure 14 is a column vector, then the x coordinate 1402 of the left-hand corner features can be estimated from the position of the first peak in the column vector, and the x coordinate, in pixels, of the right-hand corner feature 1404 can be estimated from the position of the last peak in the column vector. The y coordinates of corner features may be similarly determined from the smoothed row vector.

[0032] Peak finding is performed using statistics gathered from the smoothed row and column vectors. The first rising and falling edge is used to find the first peak while the last rising and falling edge is used to find the last peak. In order to locate these edges, an appropriate threshold is used. This threshold can be set using statistics of the smoothed row and column vectors such as the median, i.e. fifty percentile, or another percentile and some multiple of the standard deviation. For example, for relatively low density arrays in which the total area of the features is less than the total area of the background, the median is a good approximation of the valleys between the peaks. An appropriate threshold in this case would be the sum of the median and one standard deviation. For higher density arrays in which the ratio of the total area of the features to the total area of the background increases, the 30th percentile may be a more appropriate estimation of the valleys.

[0033] Additional image processing techniques may be employed to circumvent various problems in finding the corner features. For example, abnormally bright or dim areas introduced in the scan due to chemical, biological, or electrical anomalies may distort the corner finding process. These bright or dim areas may be large or small, thus creating hard-to-predict problems for the corner finding algorithm. To alleviate the impact of small area anomalies, a median filter can be used on the row and column vectors with a kernel sized large enough to eliminate significantly sized spikes, but small enough to allow peaks corresponding to features to remain. To alleviate the impact of large area anomalies, boxplot analysis or similar statistical analysis can be used to set abnormally high or low values to a reasonable number. For example, using the bloxplot method, the limits of the whiskers of the bloxplot can be used to constrain the values of data points of smoothed row and column vectors between a maximum and minimum value. Statistics for peak finding can then be recalculated based on the constrained-value data points.

[0034] Furthermore, information about the design of the array can be incorporated into the corner finding step. Information such as the number of rows and columns of features, the expected feature size, and the distances between features can be used to make the corner finding algorithms more robust. Peak finding algorithms can be iterated, each time changing the threshold value until the corners found meet the design specifications. Additionally, the peak finding algorithm can be made more exhaustive to find various numbers of peaks and to find combinations compatible with the design information.

[0035] The technique illustrated in Figure 14 is especially suitable for printed molecular arrays, in which features are relatively well spaced apart within the molecular array, as shown in Figure 1. In other types of molecular arrays, the features may be much more densely packed. In densely packed arrays having disk-shaped features, with features packed together in a two-dimensional version of the familiar closest packing lattice in which oranges are stacked in display counters, each internal feature may touch, or nearly touch, six nearest-neighbor features. For such densely-packed molecular arrays, column and row vectors may be calculated as described above, although the angle between the directions along which pixel intensity values are summed may be different than the 90° angle in the rectilinear case. For example, angles of 30°, 60°, 120°, etc. may be used for hexagonal closest-packed arrays.

[0036] Figure 15 illustrates a corner feature estimation technique more suitable for densely-packed molecular arrays for which row and column vectors have broader, less distinct peaks and valleys. In Figure 15, the row vector has broader and less distinct peaks than the row vector illustrated in Figure 14. In this case, the x coordinates of corner features can be estimated by detecting the first rising edge 1502 and last falling edge 1504 of the column vector, and then estimating the positions of the centers of the corner features as the mid points 1506-1507 of the known feature sizes 1508 and 1510 extending to the right from the first rising edge and to the left from the last falling edge. The y coordinates of corner features can be similarly determined from a smoothed row vector.

[0037] Once the x and y coordinates, in units of pixels, of the positions of the corner features are estimated from row and column vectors, as illustrated above, regions of a scanned image corresponding to the corner features can be further analyzed to refine the estimated positions of the corner features. One technique for refining the positions of features is called "blob analysis." Blob analysis comprises the analysis of pixels within a region of interest encompassing the estimated position of a feature in order to first determine a threshold pixel intensity value and to then create a binary image in which all pixels having pixel intensity values greater than the threshold value are assigned the value "1," and all pixels having pixel intensity values in the region of interest less than the threshold value are assigned the value "0." The coordinates of the centroid of the connected collection of pixels closest to the center of the region of interest in the binary image is then taken to be the refined pixel coordinates corresponding to the center of the feature.

[0038] Figures 16-17 illustrate one method for threshold determination used in blob analysis. Figure 16 illustrates an inner region of interest. In Figure 16, a square area of the scanned image centered at the estimated position of a feature

is shown. An elliptical area 1602 having a minimum size that totally encompasses the feature is centered within the area of the scanned image of the feature shown in Figure 16. This elliptical area is called an "inner region of interest." A threshold for the inner region of interest is determined by creating a histogram of the various pixel intensity values occurring within the inner region of interest. Figure 17 shows a histogram of pixel intensity values within the inner region of interest illustrated in Figure 16. A threshold pixel intensity value is determined from the histogram by proceeding from the right-most column of the histogram 1702 towards the left-most column of the histogram 1704 until a threshold point in the distribution of pixels is reached. In the illustrated embodiment, the threshold point is predetermined to be the 33rd percentile point. The threshold point may be optimized according to characteristics of the features and the arrays being analyzed, including size, detected noise levels, and measured positioning errors. The pixel intensity value at that point on the x-axis of the histogram is then taken to be the threshold value. In the case of the inner region of interest shown in Figure 16, pixel intensity values range from 0 to 9, and so the histogram of Figure 17 has ten vertical columns representing pixel intensity values 0 to 9. The number of pixels with a given intensity value is represented by the height of the corresponding column. The point on the x-axis 1704 of the histogram in Figure 17 at which 2/3 of the pixels occur to the right and 1/3 of the pixels occur to the left falls within the column of the histogram 1706 corresponding to a pixel intensity value of 1. Thus, the threshold value of 1 is used to create a binary map for the inner region of interest of Figure 16.

[0039] Figures 18 and 19 illustrate a second technique for choosing a threshold for blob analysis. Figure 18 illustrates an outer region of interest for the same area of the scanned image as shown in Figure 16. The outer region of interest is the maximally-sized elliptical area that will fit within the rectangular portion of a scanned image overlying and centered on a particular feature. The median pixel intensity for pixels within the outer region of interest ("OuterROI"), and the standard deviation for the pixel intensities of pixels within the OuterROI, are both calculated for the OuterROI, and a threshold is chosen by:

$$\text{threshold} = \text{median of OuterROI} + (A * \text{standard deviation})$$

where A is approximately equal to 1.4. Applying the above method for calculating a threshold produces a threshold value of "9" for the OuterROI illustrated in Figure 18. Figure 19 shows the binary image produced using a threshold of 9 from the rectangular portion of the scanned molecular array image shown in Figure 18. In Figure 19, all pixels having a pixel intensity in the original scanned image equal to or greater than nine are darkened, or filled in, and all other pixels are blank. The darkened pixels correspond to pixels having an intensity value of 9, and the blank pixels correspond to pixels having a value less than 9 in the binary image. The average or mean of pixel values may used in place of the median pixel value in the above equation in order to generate alternative threshold values.

[0040] A variation of this second method of determining thresholds is to use median statistics, which are more robust to outlier pixels than statistics using mean and standard deviations. A more robust estimator of standard deviation is either the inter-quartile range ("IQR"), where IQR = (75^{th} percentile - 25^{th} percentile) or the median absolute deviation ("MAD") from the median statistic metrics, which are converted to a standard deviation equivalent by dividing by a known constant (e.g. 1.35 for IQR and 0.675 for MAD). For example, the threshold may be calculated as:

$$\text{threshold} = \text{median of OuterROI} + (A * \text{standard deviation-equivalent})$$

[0041] Whether the binary image is produced according to the technique illustrated in Figures 16 and 17 or according to the technique illustrated in Figures 18 and 19, the resulting binary image is processed similarly to generate a refined pixel coordinate position for the features. The binary image is scanned to locate sets of contiguous pixels having a value of "1," or, in other words, blobs. Blob 1902 in Figure 19 is the blob containing the largest number of pixels in the binary image and is closest to the center represented by Figure 19. Then, the centroid of the blob greater than a threshold size closest to the center of the binary image is computationally determined by a method illustrated in the final subsection of the application. Figure 20 illustrates the centroid pixel of blob 1902 of Figure 19. In Figure 20, the centroid pixel 2002 is shown as a darkened pixel. The coordinates of the centroid pixel within the scanned image are then taken as the refined coordinates for the feature.

Computation of an Initial Feature Position Coordinate Grid and Refinement of the Feature Position Coordinate Grid

[0042] Using the refined corner feature positions, as determined by the techniques described in the previous subsection, an initial rectilinear feature coordinate grid can be estimated from the positions of the corner features and the known inter-feature spacings of the molecular array. Figure 21 illustrates an initial feature coordinate grid overlaying

the scanned image of a molecular array, where th units of th *x*-axis 2102 and th units of the *y*-axis 2104 are pixels. As can b s en in Figure 21, many of the features are positioned relatively closely to the intersection of f ature coordinat grid lines, but do not xactly correspond to the intersections. Thus, a technique is required to refine the initial feature coordinate grid so that f atures obs rved in th scanned image corr spond as closely as possible to the intersections of feature coordinate grid lines. In this and following discussions, it is assumed that the molecular array is optically scanned in both green and red wavelengths and includes separate green and red intensity values for each pixel. In different systems, more than two signals may be produced during analysis of the molecular array, and in such systems, all the various signals are used in the techniques to be described below.

[0043] After computing the initial feature coordinate grid, the different signals for each feature are processed in order to select strong features and to then refine the initial feature coordinate grid based on the positions of the strong features. For each feature, blob analysis is conducted, as discussed above with reference to Figures 16-20, to produce refined pixel coordinates for the center of the feature for each signal along with the size of the blob used to produce the refined coordinates. The initial position of the feature is estimated from the initial feature coordinate grid. If, based on the original feature coordinate grid, the refined position for the feature is less than some constant distance from the estimated feature position and the size of the blob determined by blob analysis is greater than a minimum threshold value and less than a maximum threshold value, then the feature is considered to be a strong feature. Otherwise, the feature is considered to be a weak feature. In Figure 21, only the strong features are shown. If a feature is strong in more than one signal, such as, in the current case, in both green and red pixel intensities, then the refined position of the feature will be calculated as the mid-point of a line segment connecting the refined positions of the feature based separately on the red and green signals.

[0044] Once the strong features are selected by the above-described technique, then the positions of the strong features are employed in a linear regression analysis to produce a refined feature coordinate grid. Figure 22 illustrates the linear regression analysis used in one embodiment of the present invention. In Figure 22, matrices are represented as rectangles, with the dimensions of the matrices indicated by numbers and letters within the rectangles. In Figure 22, the value "M" is equal to the total number of features within a molecular array and the value "N" is equal to the number of strong features found within the molecular array. The row and column spatial indices of these strong features are contained in a N x 3 strong matrix 2202. The indices of strong features are contained in two columns of the strong matrix. A third column in the strong matrix 2202 has a place holder value of "1." Thus, the first row in the strong matrix 2202 contains a first spatial index for each strong feature, the second row in the strong matrix 2202 contains a second spatial index for each strong feature, and the third row in the strong matrix 2202 contains the value "1" for each strong feature.

[0045] A second matrix employed in the linear regression analysis is the pseudo-inverse strong matrix 2204. This is an N x 3 matrix. A pseudo-inverse matrix A^+ is derived from a regular matrix A containing real number elements via singular value decomposition. For a matrix $A \in R^{M \times N}$ $(M \geq N)$with Rank (A) = N there are matrices:

$$U \in R^{M \times N}$$

$$\Sigma \in R^{M \times N}$$

$$V \in R^{M \times N}$$

such that:

$$U^T U = I_N$$

$$V^T V = I_N$$

$$A = U \Sigma V^T$$

Here I_N is the identity matrix in $R^{N \times N}$. The matrix Σ is a diagonal matrix with non-zero diagonal entries having decreasing values from the upper left corner, $\Sigma_{(1,1)}$, to the lower right corn r, $\Sigma_{(N,N)}$.

[0046] The pseudo-inverse matrix for A, denoted by A^+ is given by

$$A^+ = V\Sigma^{-1}U^T \in R^{N \times M}$$

[0047] The important property of A^+, and the one used in the grid fit regression process, is that:

$$A^+ A = I_N$$

[0048] The reference book "Matrix Computations" by H. Golub and C. F. Van Loan, Johns Hopkins University Press, 1996 (3rd edition), can be consulted for details on application and properties of the singular value decomposition (SVD) $A = U\Sigma V^T$ and the pseudo-inverse.

[0049] A third matrix used in the linear regression analysis is the centroid matrix 2206. This is an N x 2 matrix, with each row containing the centroid of a strong feature, as determined by blob analysis, in pixel coordinates. A fourth matrix used in the linear regression analysis is a feature indices matrix 2208. The feature indices matrix is a 3 x M matrix containing the indices of all features in the molecular array, and is similar in format to the strong matrix 2202. Like the strong matrix, the final row of the features indices matrix 2208 contains M instances of the value "1."

[0050] The result of linear regression analysis is a fit position matrix 2210. This is an M x 2 matrix containing the two-dimensional pixel coordinates of each feature in the molecular array resulting from linear regression analysis.

[0051] The linear regression analysis comprises two matrix multiplications. In the first matrix multiplication, the transpose 2212 of the pseudo-inverse strong matrix 2204 is multiplied by the centroid matrix 2206 to produce a 3 x 2 coefficient matrix 2214. In the second matrix multiplication, the transpose 2216 of the feature indices matrix 2208 is multiplied by the coefficient matrix 2214 to produce the fit position matrix 2210.

[0052] The grid fitting process entails an additional procedure for applying a linear regression on all of the positions for all strong features simultaneously to transform the set of grid positions of strong features to a uniformly spaced grid that better fits the data and that can provide good estimates of the positions of the weak features. The relationship between the positions, X, of a uniformly spaced set of features and the set of indices, J, for that same set of features is given by:

$$X=JC$$

where C is a 3x2 transformation matrix that contains orientation information pertaining to the rotation and offsets of the grid. Consider X to include only the positions of the N strong features and let J represent the corresponding indices. X is an Nx2 matrix, where the i^{th} row holds the position of the i^{th} feature. In practice, the strong feature positions, X_{meas}, are measured based on their centroids, and these positions deviate slightly from the theoretical grid positions, X. Moreover, X_{meas} is not uniformly spaced. To estimate the transformation matrix C to be applied in the linear regression, the above equation is rewritten as:

$$X_{meas} \cong JC$$

[0053] A least squares approximate solution for C can be found by applying the pseudo-inverse of J, as follows:

$$C \cong \text{pinv}(J)X_{meas}$$

[0054] Once this equation is solved for C, the transformation may be applied to any set of features, such as G, the index matrix for the set of all features in the grid. This gives us H, the positions of all the uniformly spaced positions of all features on the grid:

$$H=G\,C$$

[0055] An example of the linear regression analysis is provided in Figures 23A-27B. Figure 23A shows a small 9 x 12 molecular array. The molecular array 2302 contains strong features shown as light-colored disks, such as strong feature 2304. Figure 23B shows th strong matrix corresponding to the 9 x 12 molecular array of Figure 23A. Th

indices of all strong features are included in th strong matrix 2308. For example, th coordinat s of the strong feature 2310 at position (0,0) ar included in the first column 2312 of strong matrix 2308 and the c ordinates of the strong f atur 2304 at position (3,8) ar included in column 2314 of strong matrix 2308. As explain d above, a third place-keeping coordinate having the value "1" is included in the third row of the strong matrix 2308 for each strong f ature. It should b noted that, although a strong feature is not shown at position (7,0) 2306, an artifact or defect within the molecular array has caused detection of a strong feature at position (7,0) 2316.

[0056] Figure 24 shows a pseudo-inverse strong matrix calculated, as described above, from the strong matrix shown in Figure 23B. Figure 25 shows the centroid matrix that includes the pixel coordinates of the centroids of the strong features of the 9 x 12 matrix shown in Figure 23A. Figure 26 shows the coefficient matrix calculated by multiplying the transpose of the pseudo-inverse matrix (Figure 24) by the centroid matrix (Figure 25). Figures 27A-27B show the fit positions matrix calculated by multiplying the transpose of the feature indices matrix (not shown for the current example) by the coefficient matrix (Figure 26). Note also that the two-dimensional coordinates for any given feature resulting from linear regression analysis can be obtained by multiplying an index vector for the feature by the coefficient matrix shown in Figure 26. For example, multiplication of the index vector (0, 0,1) by the coefficient matrix shown in Figure 26 results in the coordinates (85.880412, 26.859797) in the first row 2702 of the fit positions matrix shown in Figure 27A-B. Similarly, multiplication of the index vectors (1,0,1) and (0,1,1) generate the coordinates in rows 2704 and 2706, respectively, of the fit positions matrix shown in Figure 27A. Note that, for very large arrays, a smaller set of features may be chosen for the linear regression analysis described above.

[0057] In subsequent signal extraction and signal variance calculations, the refined positions of the strong features, as determined by blob analysis for each signal, are used for calculations of the strong features and their respective local background regions, whereas the fitted positions enumerated in the fit positions matrix are used for weak features and their respective local backgrounds. Thus, the results of the linear regression analysis are only applied to weak features and their respective local backgrounds.

[0058] In the above case, the fitting of the grid is constrained to maintain a parallelogram symmetry. This is useful in implementations where systematic linear distance errors are small, which is generally true for linear encoders. There may be other situations where systematic distortions may break the parallel symmetry, for example, distortions arising from imaging optics where the array is imaged slightly out of the image plane. In some cases, the fit may preserve quadrangular symmetry whereas, in other cases, nonlinear fitting procedures may be required.

[0059] The array of features can be broken up into a number of smaller regions or zones, and each zone can be fitted to its own local grid. This can serve several purposes. For example, if there are known to be systematic positioning errors in certain regions, then each of those regions may be fit to a small localized grid. This method may be used for cDNA arrays and whole oligonucleotide deposition programs where sets of spots are deposited by different pins, pens, jets, or nozzles. Some ink-jet nozzles fire at fixed angles relative to other nozzles. In other deposition techniques, features are deposited onto flexible membranes. The membranes can stretch locally causing small regional distortions, making it impossible to adequately fit the whole array of features by a single grid. In this case, the grid can be broken down into a number of smaller grids, each grid with at least a few strong features. This case may better be addressed by a set of localized grids of quadrangular symmetries. When there are systematic errors associated with either pins or an ink-jet nozzle, it is not necessary for these subgrids to be non-overlapping. In fact, these independent grids, or sets of disjoint features, may overlap completely. The values of strong features associated with each independent grid are set to "1," and all other feature values are set to zero, and the grids care then independently fit.

Rejection of Outlier Pixels and Multi-Color Scans

[0060] Once feature positions are determined, whether from blob analysis for strong features or from linear regression analysis for weak features, a set of pixels from each feature is then selected for signal extraction. The selected pixels for a feature initially comprise those pixels having pixel intensity values for each signal and, optionally, for ratios of all pairs of signals, that fall within acceptable ranges within a selected region corresponding to the feature. Selection of a region for initial pixel selection for a feature can be made on the basis of geometry, e.g. selecting pixels within an ellipsoid of a size and orientation expected to include signal-bearing pixels, or may alternatively be accomplished through morphological analysis of features using image processing techniques. Selection of a set of pixels for signal extraction for a feature may be accomplished by removing, or disregarding, outlier pixels from the initial set of pixels within the selected region. Removal of outlier pixels from a feature, in one embodiment of the present invention, occurs through the following process: (1) construction of an initial inlier binary mask in which pixels of the feature have corresponding values of either 1 or 0, and all pixels within the selected region are initialized to have value 1, while all other pixels are initialized to have the value 0; (2) for each signal, identification of any outlier pixels within the selected region having intensity values outside of an acceptable range and setting of the corresponding values for those outlier pixels to 0 in the inlier mask; (3) for each possible pair of signals, identification of any pix ls within the selected region for which the ratio of th intensity values for the signals of the pair of signals fall outside of an acceptabl rang and setting

of the corresponding values in th inlier binary mask to 0; and (4) s lection of those pix ls whos corr sponding values in th inlier mask hav the valu 1 as th signal extraction pixels for the f ature.

[0061] Figur 28 illustrates th abov -described techniqu for outlier rejection. In Figure 28, plane 2802 represents th gr en signal int nsity values for a region of int rest surrounding a f atur , plane 2804 r presents the red signal intensity values for the same r gion of interest, plane 2806 represents th ratio of the gr en t red signal int nsity values, and plane 2808 represents the inlier mask corresponding to the region of interest. The process of outlier rejection can be considered as a process of setting the value of a pixel in the inlier mask 2808 to "1," if that pixel falls within acceptable ranges of intensity values in the green signal plane 2802 and red signal plane 2804 as well as within acceptable range of values of green to red signal ratios in the green to red signal ratio plane 2806. Acceptable ranges can be based on the standard deviation of the pixels comprising the feature. For example, the low limit can be set as the average of the feature minus two standard deviations and the high limit can correspondingly be set as the average of the feature plus two standard deviations. In Figure 28, the pixels having acceptable green signal values occur in the darkened regions 2812-2814, pixels having acceptable red intensity values fall within the regions of the red signal plane 2815-2817, and pixels having acceptable green signal to red signal ratios fall within regions 2818-2820 of the green to red signal ratio plane 2806. Thus, for example, pixel 2822 has the value 1 in the inlier mask 2808 because it occurs within acceptable ranges in the green to red signal ratio plane 2806, in the red signal plane 2804, and in the green signal plane 2802.

[0062] An iteration step can be added here. For example, after pixel outlier analysis, all strong features can be again subjected to blob analysis. This is useful for features with small areas of signal that are significant with respect to the threshold, but are, in fact, artifacts. In such cases, a truly weak feature may be labeled strong and the centroid of that feature may be incorrectly positioned or, alternatively, the centroid of a strong feature including an artifact may be incorrectly positioned. pulled from the true grid. Subsequent pixel outlier analysis will remove much of the original high signal outlier pixels of such features that correspond to artifacts. If a second round of blob analysis is performed, a weak feature incorrectly labeled strong may no longer have a large enough blob over threshold to be labeled strong. The feature will then be labeled as weak and its centroid will be positioned via the above linear regression analysis. In the case of strong features with artifacts, a second round on blob analysis will likely better position the corresponding centroids. Instead of performing the second round of blob analysis on all strong features, the analysis can be limited to those strong features having a blob size lower than a set limit, that is, to those features which are "borderline" strong.

Signal Extraction from Features and Propagation of Errors

[0063] For many types of analysis, it is desirable to extract the ratio of background subtracted and normalized signal intensities from the region of each feature determined according to the technique described in the previous sub-section. In this subsection, a technique for extracting signals, signal ratios, and determining variances is illustrated and described. In this subsection, the green signal intensity values for a feature are shown in Figure 29 and the red intensity values for the same feature are shown in Figure 30. The green intensity values range from 5 to 99, and the red intensity values range from 1 to 46. The average value for the green signal intensities for the feature is calculated as follows:

$$\mu_g = \frac{1}{N}\sum_{i=0}^{N-1} g_i$$

where μ_g is the average green signal, N is the number of pixels within the feature, and g_i is the green signal intensity of the i^{th} pixel. The average red signal intensity value μ_r is similarly calculated. The variance of the green signal for the feature is calculated as follows:

$$\sigma^2_g = \frac{1}{N-1}\sum_{i=0}^{N-1}(g_i - \mu_g)^2$$

[0064] The variance for the red signal intensities σ_r^2 is similarly calculated. The covariance of the red and green signal intensities for the feature is calculated as follows:

$$\sigma_{gr}{}^2 = \frac{1}{N-1}\sum_{i=0}^{N-1}(g_i - \mu_g)(r_i - \mu_r)$$

[0065] The estimated variance on the mean μ_g is calculated as:

$$\sigma_{\mu g}^2 = \frac{\sigma_g^2}{N}$$

and the estimated variance of the mean μ_r is similarly calculated. Finally, the estimated covariance on the mean is calculated as:

$$\sigma_{\mu gr}^2 = \frac{\sigma_{gr}^2}{N}$$

[0066] The values of these parameters for the example features of Figure 29 and Figure 30 are provided below:

$$N = 60$$

$$\mu_g = 75.73$$

$$\mu_r = 29.93$$

$$\sigma_g^2 = 930.94$$

$$\sigma_r^2 = 162.47$$

$$\sigma_{gr}^2 = 323.00$$

$$\sigma_{\mu g}^2 = 15.51$$

$$\sigma_{\mu gr}^2 = 5.38$$

$$\sigma_{\mu gr}^2 = 5.38$$

$$\sigma_{\mu r}^2 = 2.71$$

[0067] These parameters represent raw signals and raw signal statistics. Continuing on with further processing, a background level must be subtracted from each feature signal due to differences in background intensities within the various color channels. There are a number of different methods to determine the background intensity for different signals. In a first method, the average background intensity for each signal can be measured and reported by the scanner. In a second method, a population of pixels in a region surrounding or in proximity to a feature, known as a local background, can be averaged to provide a different background measurement for each feature. In a third method, the background for a signal can be computed as the average of a collection of local backgrounds. For example, the nine nearest neighboring local backgrounds or, all the local backgrounds can be pooled together to obtain a background value. In a fourth method, the background can be estimated as the average signal intensity of a minimum feature or minimum local background in the molecular array. Finally, in a fifth method, the background can be calculated as the average of all or a subset of appropriate control features included in the molecular array. The appropriate control features can comprise a negative control designed not to hybridize with any target sample molecules or a deletion control designed to estimat the amount of non-specific hybridization of a corresponding perfect match feature. Alter-

native median statistics and robust standard deviation estimators can be substituted for the use of mean and standard deviation in all the above examples of background calculations and in subsequent discussions of error propagation. Figure 31 illustrates a number of local backgrounds. In Figure 31, local backgrounds 3102-3110 surround features 3112-3120. Figure 32 illustrates a minimum feature. In Figure 32, feature 3201 has the lowest intensity for a particular signal. Thus, the average pixel intensity values for that signal of feature 3201 is taken as the background for that signal.

[0068] Alternative robust statistical metrics can be used in the above pixel outlier, feature extraction, and background calculation algorithms and in subsequent discussions of error propagation. Examples of a robust location metric is the median or the trimmed mean. Examples of robust dispersion metrics are the IQR and MAD metrics discussed above in the threshold determination section. Two of the advantages of using robust metrics, as opposed to the use of average and standard deviation, are that they are far less influenced by outliers and they do not make assumptions of the nature of the underlying distribution of data.

[0069] The background averages and variances associated with each local background are notationally represented as $\mu_{bg,g}$, $\mu_{bg,r}$, $\sigma_{\mu bg,g}^2$, $\sigma_{\mu bg,r}^2$. These parameters can be computed similarly to the computation for the averages and variances of the raw signals, described above. However, if the background is estimated from a collection of local backgrounds or a subset of control features, and thus represents an average of averages, or pooled average, then variance of the background measurement may be properly represented as follows. Let $\mu_{BG,g}$ and $\mu_{BG,r}$ represent the average background signal for the green and red features respectively. Let $\sigma_{BG,g}^2$ and $\sigma_{BG,g}^2$ represent the estimated variance on $\mu_{BG,g}$ and $\mu_{BG,r}$ respectively. Method 3, outlined above, is treated below, where the pooled average is:

$$\mu_{BG,g} = \frac{\sum_{i=0}^{L-1}\left(\mu_{bg,g,i} \Big/ \sigma^2_{\mu bg,g,i}\right)}{\sum_{i=0}^{L-1}\left(1 \Big/ \sigma^2_{\mu bg,g,i}\right)}$$

and the variance of the average of averages is calculated as follows:

$$\sigma_{BG,g}^2 = \frac{1}{\sum_{i=0}^{L-1}\left(\frac{1}{1/\sigma_{\mu bg,g,i}}\right)}$$

where L is the number of local backgrounds used in calculating the average, for example, the nine nearest neighboring local backgrounds.

[0070] The background-subtracted green signal μ_G is calculated as follows:

$$\mu_G = \mu_g - \mu_{BG,g}$$

and the background-subtracted red signal μ_R is similarly calculated. The variance of the background subtracted green signal μ_G^2 is calculated as follows:

$$\sigma_G^2 = \sigma_{\mu g}^2 + \sigma_{BG,g}^2$$

and the variance of μ_R is similarly calculated. The covariance of the background-subtracted green and red signals σ^2_{GR} is calculated as follows:

$$\sigma_{GR}^2 = \sigma_{\mu gr}^2$$

[0071] To facilitate biological interpretation and downstream analysis of the data, the statistical significance of feature signals needs to be determined. A problem arises if, for example, the red channel signal and green channel signal of the same feature are both indiscernible from their surrounding local background, but the green channel signal is still

twice as bright as the red channel signal. The user, in this case, may obtain a false r sult indicating a two-fold increase in xpression by the gre n chann l If th ratios are calculated with data that is not significantly differ nt compar d to a blank, where the blank is on of the background options outlined above. This problem may b address d by performing statistical significance t sts on f ature data. A two-tail d student's t-test is p rform d on th population of pixels comprising th feature with the appropriate population comprising the background signal. The population used for the background signal depends on the method chosen for background subtraction. As discussed below, this significance information is used when calculating the log of the ratio of one color channel signal to another color channel signal of the same feature.

[0072] As can be seen in the examples feature signal intensities of Figures 29 and 30, the average value for one signal intensity for a particular feature may be quite different from the average value for a different signal for the same feature. For example, the labeling and photon-efficiencies of different dyes, or chromophores, may be different. In Figures 29 and 30, $\mu_g = 75.73$ and $\mu_r = 29.93$. In this case, in order to compare the green and red signals of the features for a molecular array, the green and red signals for all features must be normalized. Signal response curves may be employed during normalization. A linear normalization curve is assumed in the following discussion. In one embodiment of the present invention, using the assumption that, on average, genes are not differentially expressed, the average log ratio of red to green background-subtracted signals is forced to be zero. This is accomplished by computing a dye normalization factor for each color channel, $DNFactor_G$ and $DNFactor_R$ Thus, using M feature signals:

$$DNFactor_G = 10^{\left(\frac{\sum_{j=1}^{M} \log_{10}(\mu_{G,j})}{M}\right)}$$

[0073] Using the dye normalization factors, all the feature signals are then normalized. Let μ_{Rn} and μ_{Gn} represent the dye normalized red and green signals.

Thus,

$$\mu_{Gn} = DNFactor_G \cdot \mu_G$$

and

$$\mu_{Rn} = DNFactor_R \cdot \mu_R$$

[0074] Let σ_{Gn}^2 and σ_{Rn}^2 represent the variance on μ_{Gn} and μ_{Rn} respectively, where, following the standard error propagation equation:

$$\sigma_{Gn}^2 = DNFactor_G^2 \cdot \sigma_G^2$$

$$\sigma_{Gn}^2 = DNFactor_R^2 \cdot \sigma_R^2$$

[0075] Let σ_{GRn}^2 represent the propagated covariance where:

$$\sigma_{GRn}^2 = DNFactor_G \cdot DNFactor_R \cdot \sigma_{GR}^2$$

[0076] Finally, let *LR* represent the $\log_{10}$(*red to green ratio*) where:

$$LR = \log_{10}\left(\frac{\mu_{Rn}}{\mu_{Gn}}\right)$$

[0077] To propagate the error at this step, the standard error propagation equation for $x = f(u,v)$ is used

$$\sigma_z^2 \cong \sigma_u^2\left(\frac{\partial x}{\partial u}\right)^2 + \sigma_v^2\left(\frac{\partial x}{\partial v}\right)^2 + 2\sigma_{uv}^2\left(\frac{\partial x}{\partial u}\right)\left(\frac{\partial x}{\partial v}\right)$$

[0078] In the current case, letting σ_{LR}^2 represent the variance of *LR*, application of the above formula yields:

$$\sigma^2_{LR} \cong \frac{1}{(\ln 10)^2}\left[\frac{\sigma_{Rn}^2}{\mu_{Rn}^2} + \frac{\sigma_{Gn}^2}{\mu_{Gn}^2} - 2\left(\frac{\sigma_{GRn}^2}{\mu_{Rn}\cdot\mu_{Gn}}\right)\right]$$

[0079] It is important at this point to test for significance of the red and green channel data. If both color channel signals for a feature are found to be insignificantly different from the population describing the feature's background, typically at a significance level < .01, then set the log ratio log(0/0) to be log(1) which is defined to be 0. This avoids the erroneous result of a gene expression level artificially high or low based on data that is considered to be essentially the same as some background level.

Implementation

[0080] In this subsection, a series of flow-control diagrams are employed to describe the implementation of routines for establishing the orientation of a rectilinear grid for indexing features of a molecular array, determining the pixel-based coordinates of the corner features of a molecular array, establishing an initial mapping between pixel coordinates and the indexing grid, and determining positions of the features using the initial mapping. The positions of weak features can then be refined according to the linear regression analysis method described above. The linear regression analysis, subsequent rejection of outlier pixels, and signal extraction from features along with propagation of errors can be straightforwardly implemented according to the mathematical descriptions provided in previous subsections.

[0081] Figures 33-38 provide a series of flow-control diagrams that describe the method of refining feature coordinates by blob analysis, described above with reference to Figures 16-20. Figure 33 is flow-control diagram for the routine "refine feature coordinates." This routine is the top-level routine for implementing refinement of feature coordinates using blob analysis, as discussed above with reference to Figures 16-20. In step 3302, the pixels in a scanned image of a molecular array within a region of interest surrounding initial coordinates for the center of a feature are processed to determine the average intensity values of pixels in the region of interest, the variance of the intensity values of pixels in the region of interest, the medium intensity value in the region of interest, and, as a by-product, the number of pixels in the region of interest. The latter quantity may be constant, or may vary slightly depending on the position of the feature within the molecular array and the orientation of an indexing grid determined for the molecular array. In step 3303, a threshold value is set using an above-described formula. In the *for*-loop comprising steps 3304-3308, a map "binmap" with elements corresponding to pixels of the scanned image of the molecular array is initialized by setting each element of the binmap equal to 1, in step 3307, if the intensity of the pixel corresponding to the binmap element is greater than the threshold value calculated in step 3303, and setting the element value to 0, in step 3308, if the intensity of the corresponding pixel value in the scanned image of the molecular array is less than or equal to the threshold value, as determine in step 3305. Next, in step 3309, the binary map "binmap" is prepared for blob analysis via a call to the routine "prepare binmap," following execution of which the routine "find equivalences" is called in step 3310 to identify all blobs within the region of interest, and, finally, in step 3311, the routine "find blob" is called to identify a sufficiently large, closest blob to the unrefined center position of the feature, refine the feature coordinates according to the centroid of this blob, and return the size, in pixels, of the blob.

[0082] Figure 34 is a flow-control diagram of the routine "prepare binmap." This routine analyzes the map "binmap," initialized in steps 3304-3308 of Figure 33, in order to set the values of elements of *binmap* initially having the value "1" to a successive set of blob numbers, where each initial blob comprises a set of contiguous pixels corresponding to *binmap* elements having the value "1." In other words, all elements of *binmap* initially having the value "1" are assigned new values that represent partitioning of corresponding pixels into groups of contiguous pixels. The variable "current" is set to 1, in step 3401, and is used as the next blob value to be assigned to a newly-found group of contiguous pixels having intensities gr ater than th threshold valu . Th r maining steps 3403-3410 compris two nested *for*-loops in which the variables "i" and "j" ar together assigned all possible index-pair values "(i,j)" for ev ry el ment in

binmap. Thus, starting with step 3404, each element in *binmap* is considered separately. If the element is not the first element in a row, and the previous element in the same row has already been assigned a blob number, then, in step 3405, the currently considered element is set to the blob number of the previous, adjoining element in the row. If, as detected in step 3406, the element is not the first element in a column, and the adjacent element preceding the element in the same column has already been assigned a blob number, then the currently considered element is assigned that blob number in step 3407. Otherwise, in step 3408, the currently considered element is assigned the next blob number, and the variable "current" is incremented. At the conclusion of execution of the routine "prepare binmap," *binmap* is partially partitioned into blobs, or equivalence classes, based on physical contiguity of high-intensity pixels within the scanned image of the molecular array.

[0083] Figure 35 illustrates the routine "find equivalences." This routine transforms the partially partitioned *binmap* into a fully partitioned *binmap*. Essentially, the routine "find equivalences" identifies pairs of contiguous blobs and coalesces them into a single blob having a blob number equal to the lowest of the blob numbers of the two blobs of the pair. Contiguous blobs may arise for irregularly shaped blobs having, for example, arms separated by elements of 0 value within the map "binmap."

[0084] In step 3501, the routine "find equivalences" initializes the array "equivalences" to contain monotonically increasing values starting with 0. Thus, *equivalences*[0] contains the value "0," *equivalences*[1] contains the value "1," etc. The array "equivalences" is indexed by blob numbers, and contains for each blob number any lower-numbered blob number identified by the routine " find equivalences" as being equivalent to the blob. In step 3502, the routine " find equivalences" sets the variable "num" to the value "0." If, following execution of the *for*-loop comprising steps 3503-3507, the variable "num" still has the value "0," as detected in step 3508, then all pairs of equivalent blobs have been identified and resolved. The *for*-loop comprising steps 3503-3507 sets the indices "i" and "j" to all possible index pairs "(i,j)" for elements in *binmap*, and the routine "set equivalence," to be described below with reference to Figure 36, is called in step 3505 for each binmap element. Once all the equivalences have been found, then, in the nested *for*-loops comprising steps 3509-3516, *binmap* elements are reassigned, where necessary, to their final partition value. In step 3511, the variable "n" is assigned to be the current value of a currently considered element of the binmap. Then, in steps 3512-3513, n is iteratively reassigned to the value stored in the array " equivalence" at index n until n equals the value stored in the array "equivalences" at index n, as detected in step 3512. Finally, in step 3514, the binmap element is assigned the value *n*.

[0085] Figure 36 is a flow-control diagram for the routine "set equivalence," called in step 3505 of Figure 35. This routine comprises a complex Boolean expression that detects whether a *binmap* element is adjacent either to a previous element in its row or a previous element in its column, and, if so, sets an equivalence relationship in the array " equivalences" to reflect the fact that the blobs to which the two considered *binmap* elements belong are equivalent. If an equivalence is detected, then the variable "num" is incremented, in step 3610, in order to cause an additional iteration of the nested *for*-loops comprising steps 3503-3507 of Figure 35. Specifically, the routine "set equivalence" first determines whether the value of the currently considered *binmap* element is greater than 0 in step 3601. If so, then, in step 3602, the routine "set equivalence" determines whether a previous binmap element in the same column as the currently considered element has a different non-zero value than the currently considered element and, if so, sets the appropriate element in the array "equivalences" to reflect the equivalence between the two blobs that contain the two elements in steps 3603-3605. Otherwise, in step 3606, the routine " set equivalence" determines whether a previous, adjacent binmap element in the same row as the currently considered binmap element has a different non-zero value than the currently considered binmap element and, if so, sets an element of the array "equivalences" to reflect the equivalence of the two blobs to which the two elements belong in steps 3607-3609.

[0086] Figure 37 is a flow-control diagram for the routine " find blob." The routine " find blob" finds the closest blob to the unrefined coordinates of the center of the feature that is greater than a threshold size, and then refines the feature coordinates to correspond to the coordinates of the centroid of the blob by calling the routine "set feature coordinates" in step 3709. First, the routine "find blob" counts the number of pixels, or *binmap* elements, in each blob and stores the counts in the array "groups" in steps 3701-3704. Then, the routine " find blob" selects a threshold value from one of the values stored in the array "groups" in step 3705. The selection of the threshold value may be based on any number of different criteria, as discussed above. Then, in a *for*-loop comprising steps 3706-3707, the routine " find blob" selects each element from *binmap*, starting with the element closest to the center of the feature, and proceeds outward through the region of interest in a spiral, determining in step 3707 whether the next element in the spiral has a value that exceeds the threshold calculated in step 3705. The first binmap element in the spiral that exceeds the threshold breaks the *for*-loop of steps 3706-3707, resulting in the variable "blob" being assigned to the value of the first *binmap* element that breaks the *for*-loop in step 3708. Thus, the variable "blob" is assigned the number of the blob closest to the unrefined center position of the feature having greater than a threshold size.

[0087] The routine " set feature coordinates," called by the routine " find blob" in step 3709, determines the centroid of the blob determined by the routine " find blob" and sets the coordinates of the center of the feature to the centroid of the blob. In step 3801, the routine " set feature coordinates" determines whether the blob found by the calling routine

" find blob" is acceptable for d t rmining a refin d position for the center of the feature. As discussed above, this determination depends on the size of the blob and distanc of the blob from th unrefined coordinates of th cent r of the featur . The routine "setf atur coordinates" makes the determination in step 3801 and, if the blob is not acc ptable, sets an indication to indicate that th f ature is weak in step 3802 and then returns. Otherwise, in steps 3803-3806, the routine "set feature coordinat s" accumulates th x and y coordinates of each lement f the blob in th variables "mx" and "my" and then, in step 3807, calculates the refined coordinates of the center of the feature as the coordinates of the centroid of the blob. In step 3808, the routine "set feature coordinates" sets an indication that the feature is strong, and then returns. In either the case that the strong or weak feature, the return value passed back by the routine " set feature coordinates" is the number of pixels or *binmap* element in the blob. The value is passed back through the routine " find blob" to the routine "refine feature coordinates," from which it is passed back to the code calling the routine " find feature coordinates."

[0088] Figure 39 is the flow-control diagram of the routine "establish orientation" that determines a correct orientation for an initial indexing grid, as discussed above with reference to Figures 4-13. In step 3901, the routine "establish orientation" receives a digitized molecular array scanned image and sets the variable "res" to 0. In the *for*-loop comprising 3902-3907, the routine "establish orientation" calculates row and column vectors, in step 3903, for each angular orientation between -5 degrees and 5 degrees, and for each angular orientation, sets the variable "tmp" to the sum of the variances of the row and column vectors in step 3904. If the value of *temp* is greater than the value of *res*, as detected by the routine "establish orientation" is step 3905, then, in step 3906, res is assigned to the value of *tmp* and the variable "a" is assigned to be the current angle. In step 3908, the routine "establish orientation" returns the angle "a" as well as the saved row and column vectors saved in step 3906.

[0089] Figure 40 is a flow-control diagram for the routine " establish corners." This routine determines the positions of corner features as discussed above with reference to Figures 14 and 15. First, in step 4001, the routine "establish corner" calls the routine " establish orientation" to set an overall orientation for a rectilinear indexing grid. Next, in step 4002, the routine "establish corners" smoothes the row and column vectors returned by the routine " establish orientation" according to the method described above with reference to Figures 6A-B through Figure 9. in step 4003, the routine "establish corners" sets the variables "low_y," "high_y," "low_x," and "high_x" to the positions of the left and right-hand peaks in the row vector and the positions of the left and right-hand peaks in the column vector, respectively. In step 4004, the routine " establish corners" sets initial pixel-based coordinates for the center of each feature according to the four variables whose values are set in the above step 4003. Finally, in the *for*-loop comprising steps 4005-4007, the routine "establish corners" calls the routine "refine feature coordinates" for each corner feature to refine the positions of each of the corner features.

[0090] Figure 41 is a flow-control diagram for the routine "establish grid." This routine establishes an initial rectilinear coordinate system, or indexing system, for a scanned molecular array based on the defined positions of feature coordinates, next determines the positions of all additional strong features, and then, in step 4113, calls a linear regression analysis routine that carries out linear regression as discussed above with reference to Figures 21-27B. First, in step 4101, the routine " establish grid" uses the refined positions of the corner features, returned by the routine "establish corners," to create a rectilinear indexing grid for all features. This indexing grid can be straightforwardly calculated from the orientation angle established by the routine "establish orientation" and by dividing the dimensions of the scanned image of the molecular array, established by the initial positions of the corner features, into equal-sized increments according to the number of features along each dimensional axis. Steps 4102-4112 together comprise a *for*-loop in which two different sets of data, or signals, are used to determine whether each feature, other than the corner features, is a strong feature for which coordinates can be used as determined, or whether the feature is a weak feature, for which coordinates must be determined via the linear regression analysis carried out in step 4113. In step 4103, the currently considered features initial coordinates are set according to the rectilinear indexing determined in step 4101. Then, the routine "refine feature coordinates" is called in step 4104 with respect to one of two data signals, assumed in the present case to be the red signal. If, as detected by the routine "establish grid" in step 4105, the currently considered feature is strong with respect to the red signal, then, in step 4106, the refined coordinates based on the red signal are saved, and the coordinates of the feature are refined via a call to the routine "refine feature coordinates" in step 4107 with respect to the green signal. If the refinement based on the green signal results in the feature being classified as strong, as detected by the routine "establish grid" in step 4108, then the refined coordinates of the feature are set to the average of the refined coordinates based on the green signal and the saved red coordinates in step 4109. Otherwise, the refined coordinates for the feature are set to the saved coordinates based on the red signal in step 4110. If the feature is not classified strong with respect to refinement related to the red coordinates, then, in step 4111, the routine "refine feature coordinates" is called with respect to the green signal. Finally, following completion of the *for*-loop comprising steps 4102-4112, a linear regression analysis is conducted, in step 4113, according to the mathematical methodology described above with reference to Figures 21-27B.

[0091] It is not intended that th invention be limited to this embodiment. Modifications will be apparent to those skilled in th art. For xample, an almost limitless number of implementations of the feature extraction method of the

present invention are possible using different programming languages, operating systems, programming styles and techniques, and different scanning devices and scanning device interfaces. When more than two different types of data signals are available, the described method can be easily extended to incorporate the additional data signals, as indicated in the above discussion. The feature extraction methodology outlined above can be employed for many different types of molecular arrays with many different sizes and styles of features placed on the surface of the molecular array in many different regular patterns. The feature extraction method outlined above is applicable to an almost limitless number of different types of molecular arrays with regard to the molecular components of the features and with regard to molecules hybridized or bound to those components during experimentation. Depending on the type of scanning device used for analyzing a molecular array, additional steps and techniques may be added to the above-described feature extraction method in order to account for variations in interfaces, hardware components, and other parameters.

[0092] Information on the array layout can be obtained from a code associated with an array. The obtained array layout information can then be used by a processor in an array scanner (or elsewhere) to construct the second pattern. Alternatively, the code may otherwise be associated with the array. The code may, for example, be a bar-code or other code affixed to the array substrate or housing, or provided to an array user in the same package as the array, or in some other manner physically associated with the array when provided to the user. In particular, all of the techniques described in co-pending U.S. patent application Serial No. 09/302,898 entitled " Polynucleotide Array Fabrication" by Caren et al. (filed April 30, 1999) and Serial No. 09/558,532 entitled "Array Fabrication with Drop Detection" by Schantz et al. (filed April 26, 2000), for providing or obtaining array error information, may be alternatively or additionally be used for providing (such as at the fabrication location) or obtaining (such as at the end user location) any other array layout information (such as the composition of features such as polynucleotide sequences, or any other feature characteristics such as features size or location) and additionally or alternatively, the second pattern or information on the construction of the second pattern. Additionally or alternatively, all of such techniques can also be used for providing or obtaining genetic information associated with a set of one or more features on one or more arrays. For example, such information may be information on genes which are associated with one or more polynucleotide features on an array (for example, an identification of genes potentially responsible for an observed signal pattern from array features). Thus, the disclosures of those applications can alternatively or additionally be read with reference to " error information" or " error maps" being replaced with any one or more of "array layout information", "second pattern information" or "genetic information associated with a set of one or more features".

[0093] The disclosures in United States patent applications no. 09/589,046 and 09/659,415, from which this application claims priority, and in the abstract accompanying this application are incorporated herein by reference.

Claims

1. A method, embodied in a computer program, for automated extraction of data from a molecular array having features arranged in a regular pattern, the method comprising:

   receiving a number of images of the molecular array, each produced by scanning the molecular array to determine intensities of data signals emanating from discrete positions on a surface of the molecular array;
   estimating initial positions of selected marker features within an image of the molecular array;
   calculating refined positions of the selected marker features within the image of the molecular array;
   using the refined positions of the selected marker features to computer an initial coordinate system for locating features of the molecular array in the number of images of the molecular array;
   using the initial coordinate system to locate positions of strong features within one or more images of the molecular array;
   refining the positions of strong features within the one or more images of the molecular array by analyzing data signal intensity values in regions of the one or more images of the molecular array that contain the strong features;
   using the refined positions of strong features in the one or more images of the molecular array to calculate a refined coordinate system to locate positions of weak features within the number of images of the molecular array;
   using the refined positions of strong features in the one or more images of the molecular array to calculate a refined coordinate system to locate positions of local background regions surrounding all strong and weak features within the number of images of the molecular array; and
   extracting data from strong features, and their respective local background regions, within the number of images of the molecular array using the refined positions of strong features within the number of images of the molecular array and extracting data from work features, and their respective local background regions, within the number of images of the molecular array using locations for the weak features calculated from the refined

coordinate system.

2. The method of claim 1 wherein data signals emanating from discrete positions on the surface of the molecular array include:

fluorescent emission from fluorophores incorporated into molecules bound to features of the molecular array;
radiation emitted by radioisotopes incorporated into molecules bound to features of the molecular array; and
light emission from chemoluminescent moieties incorporated into molecules bound to features of the molecular array.

3. The method of claim 1 wherein each image of the number of images comprises an array of pixels, each pixel having a data signal intensity value.

4. The method of claim 3 wherein the features of the molecular array are arranged in a rectilinear grid, wherein corner features are selected as marker features, and wherein estimating initial positions of selected marker features within an image of the molecular array includes:

calculating row and column vectors by considering the values of pixels in rows and columns of the image;
determining a first and last peak in the row and column vectors; and
using pixel coordinates of the first and last peaks in the row vector to determine horizontal coordinates of the corner features and using pixel coordinates of the first and last peaks in the column vector to determine vertical coordinates of the corner features.

5. The method of claim 4 wherein row and column vectors are calculated for various orientations of an image produced by scanning the molecular array, and the orientation at which the column and row vectors have sharpest and most distinct peaks is selected as the orientation for subsequent calculations.

6. The method of claim 3 wherein calculating refined positions of the selected marker features within the image of the molecular array includes:
for each selected marker feature,

creating a binary image of the selected marker feature; and
using blob analysis to determine refined pixel coordinates for the feature.

7. The method of claim 6 wherein the binary image is generated by:

creating a histogram of intensity values within an inner region of interest that contains pixels corresponding to the feature;
selecting an intensity value as a threshold value from the histogram based on a ratio of a number of pixels having an intensity value greater than the threshold value divided by a total number of pixels within the inner region of interest; and
for each pixel in the binary image, setting the value of the pixel to 1 if the intensity value of a corresponding pixel in the image is greater than or equal to the threshold value, and setting the value of the pixel to 0 if the intensity value of the corresponding pixel in the image is less than the threshold value.

8. The method of claim 6 wherein the binary image is generated by:

determining a median intensity value and standard deviation of intensity values of pixels within an outer region of interest that contains the pixels corresponding to the feature;
selecting as a threshold value an intensity value that is a number of standard deviations greater than the median intensity value;
for each pixel in the binary image, setting the value of the pixel to 1 if the intensity value of a corresponding pixel in the image is greater than or equal to the threshold value, and setting the value of the pixel to 0 if the intensity value of the corresponding pixel in the image is less than the threshold value.

9. The method of claim 3 wherein a strong feature is a feature for which a largest blob produced during blob analysis of the feature is greater than or equal to a minimal size and less than a maximum size and located within a threshold distance from a center position of th featur estimated from the initial coordinate system.

10. The method of claim 9 wher in blob analysis is used to refine the positions of strong featur s and their respectiv local background regions within th on or mor images of the molecular array by analyzing data signal values in ragions of th on or more images of th molecular array that contain the strong f atures.

11. The method of claim 3 wherein linear r gr ssion analysis is used to calculate a refined coordinat system to locate positions of weak features and their respective local background regions within the number of images of the molecular array.

12. The method of claim 11 wherein the linear regression analysis is carried out using two matrix multiplication steps, in a first step calculating a coefficient matrix by multiplying together a transpose of a pseudo-inverse of a strong matrix containing indices of strong features with a centroid matrix containing coordinates of centroids of strong features, and in the second step calculating a fit positions matrix by multiplying together a transposed feature indices matrix and the coefficient matrix calculated in the first step.

13. The method of claim 1 wherein extracting data from strong features, and their respective local background regions, within the number of images of the molecular array using the refined positions of strong features within the number of images of the molecular array and extracting data from weak features, and their respective local background regions, within the number of images of the molecular array using locations for the weak features calculated from the refined coordinate system includes:

determining which pixels are included in the extraction of signal from each feature or local background region using statistical methods for pixel outlier identification;
determining averages and variances of data signal intensities for features of the molecular array and covariances for one or more pairs of data signal intensities;
determining averages and variances for background data signal intensities;
determining background-subtracted averages and variances of data signal intensities for features of the molecular array and background-subtracted covariances for one or more pairs of data signal intensities;
normalizing the data signal intensities, averages, and variances; and
calculating ratios and variances of ratios of pairs of normalized data intensity signals.

14. A system for automated extraction of data from a molecular array having features arranged in a regular pattern, the system comprising:

a scanning component that produces images of the molecular array representing intensities of data signals emitted from discrete positions on a surface of the molecular array;
a computer program that processes the images of the molecular array produced by the scanning component to index features in the images of the molecular array corresponding to molecules bound to features of the molecular array and that extracts data from the indexed features within images of the molecular array; and
a computer for executing the computer program.

15. The system of claim 14 wherein data signal intensities emanating from discrete positions on the surface of the molecular array include:

radiation emitted by radioisotopes incorporated into molecules bound to features of the molecular array;
fluorescent emission from fluorophores incorporated into molecules bound to features of the molecular array; and/or
light emission from chemoluminescent moieties incorporated into molecules bound to features of the molecular array.

16. The system of claim 14 wherein the computer program processes the images of the molecular array and extracts data from indexed features within images of the molecular array by:

receiving a number of images of the molecular array produced by the scanning component;
estimating initial positions of selected marker features within an image of the molecular array;
calculating refined positions of the selected marker features within the image of the molecular array;
using the refined positions of the selected marker features to compute an initial coordinate system for locating features of th mol ular array in the number of images of the molecular array;
using the initial coordinat system to locat positions of strong features within one or mor images of the

molecular array;
refining th positions of strong f atures within the on or mor images of th molecular array by analyzing data signal intensity values in regions of th one or mor images of the molecular array that contain the strong features;
using th r fined positions of strong features in the on or more images of the molecular array to calculate a refined coordinate system to locate positions of weak features within the number of images of the molecular array;
using the refined positions of strong features in the one or more images of the molecular array to calculate a refined coordinate system to locate positions of local background regions surrounding all strong and weak features within the number of images of the molecular array; and
extracting data from strong features, and their respective local background regions, within the number of images of the molecular array using the refined positions of strong features within the number of images of the molecular array and extracting data from weak features, and their respective local background regions, within the number of images of the molecular array using locations for the weak features calculated from the refined coordinate system.

17. The system of claim 16 wherein the computer program calculates background-subtracted averages, background-subtracted variances, and background-subtracted confidence intervals for data signal intensities integrated over features in the images corresponding to features of the molecular array.

18. The system of claim 16 wherein the computer program calculates background-subtracted averages, background-subtracted variances, and background-subtracted confidence intervals for ratios of pairs of data signal intensities integrated over features in the images corresponding to features of the molecular array.

19. A method of evaluating an orientation of a molecular array having features arranged in a pattern, including the steps of:

(a) receiving an image of the molecular array produced by scanning the molecular array to determine data signals emanating from discrete positions on a surface of the molecular array;
(b) calculating an actual result of a function on pixels of the image lying in a second pattern; and
(c) comparing the result of step (b) with an expected result which would be obtained if the second pattern had a predetermined orientation on the array.

20. A method according to claim 19, wherein:

the features of the molecular array are arranged in a rectilinear grid and the pattern comprises a rectilinear grid of rows and columns; and
step (b) comprises calculating row and column vectors by summing pixels in the rows and columns.

21. A method according to claim 19 or 20, including, when the results of the comparison in step (c) are outside a predetermined difference, the steps of altering the orientation of the second pattern on the array and repeating steps (b) and (c) until the results of the comparison are within the predetermined difference.

22. A method according to claim 19, 20 or 21, including the steps of:
obtaining information on the array layout using a code associated with the array and constructing the second pattern based on the obtained array layout information.

23. A method according to claim 22, wherein the array layout information is obtained from a remote location.

24. A method according to any of claims 19 to 23, wherein the orientation comprises rotational orientation.

25. A system for evaluating an orientation of a molecular array having features arranged in a pattern, including:

(a) means for receiving an image of the molecular array produced by scanning the molecular array to determine data signals emanating from discrete positions on a surface of the molecular array;
(b) calculating means for calculating an actual result of a function on pixels of the image lying in a second pattern; and
(c) comparing means for comparing th calculated result with an xpect d result which would be obtained if

the second pattern had a predet rmined orientation on the array.

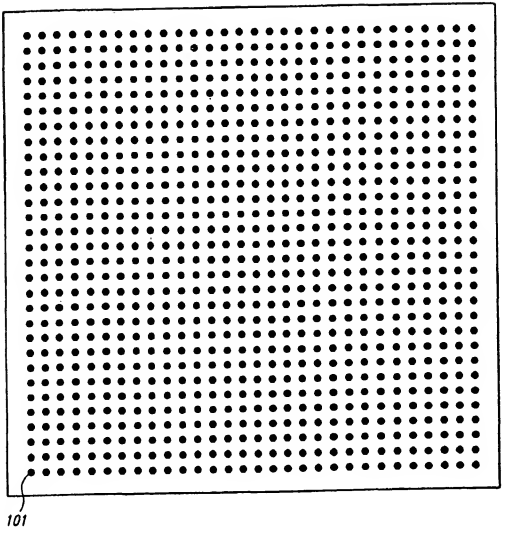


Fig. 1

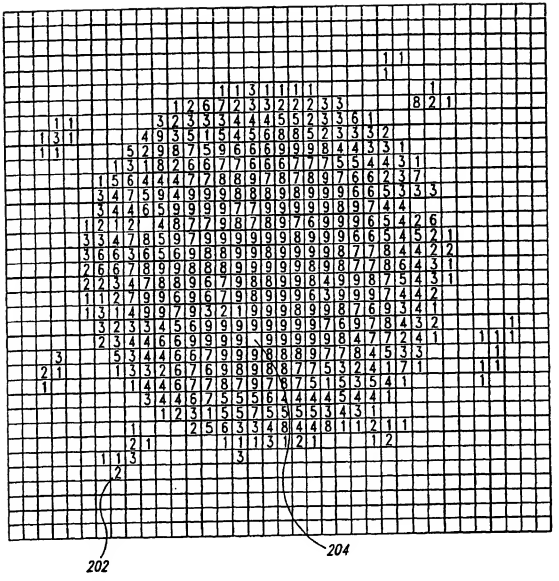


Fig. 2

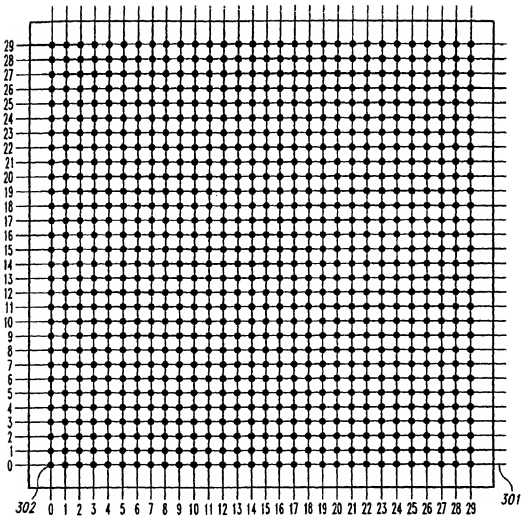

29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
302
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
301

Fig. 3

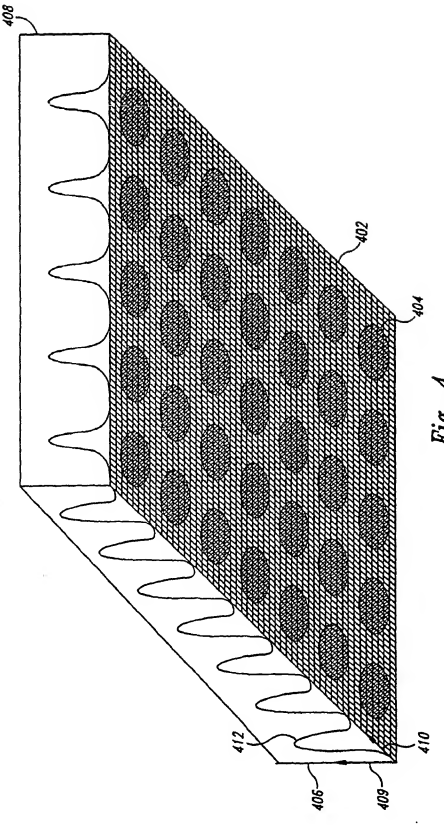


Fig. 4

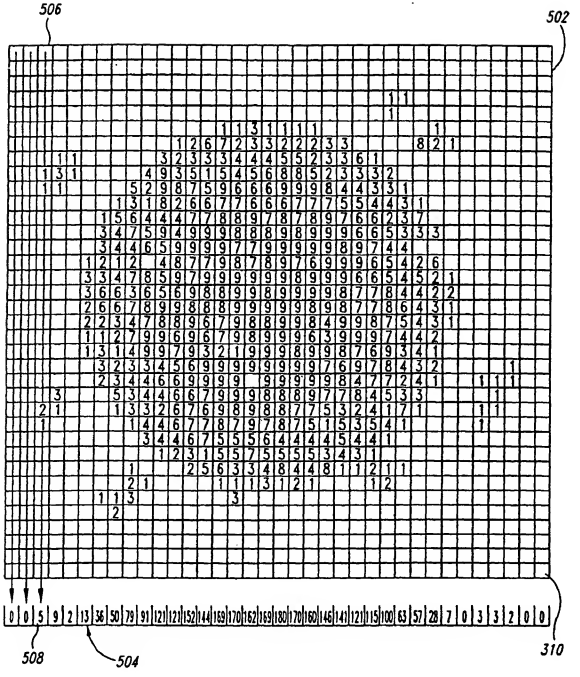
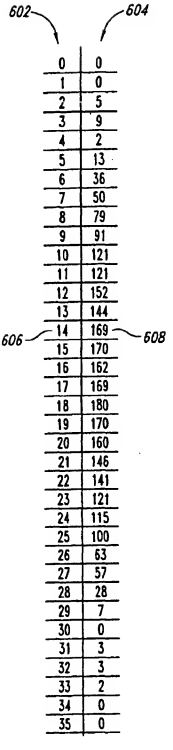
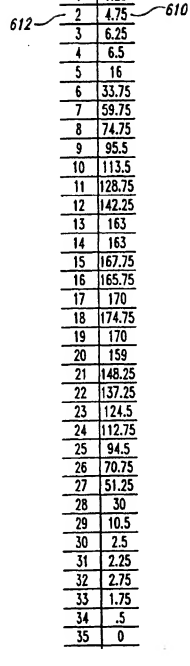


Fig. 5



0	0
1	0
2	5
3	9
4	2
5	13
6	36
7	50
8	79
9	91
10	121
11	121
12	152
13	144
14	169
15	170
16	162
17	169
18	180
19	170
20	160
21	146
22	141
23	121
24	115
25	100
26	63
27	57
28	28
29	7
30	0
31	3
32	3
33	2
34	0
35	0

Fig. 6A



0	0
1	1.25
2	4.75
3	6.25
4	6.5
5	16
6	33.75
7	59.75
8	74.75
9	95.5
10	113.5
11	128.75
12	142.25
13	163
14	163
15	167.75
16	165.75
17	170
18	174.75
19	170
20	159
21	148.25
22	137.25
23	124.5
24	112.75
25	94.5
26	70.75
27	51.25
28	30
29	10.5
30	2.5
31	2.25
32	2.75
33	1.75
34	.5
35	0

Fig. 6B

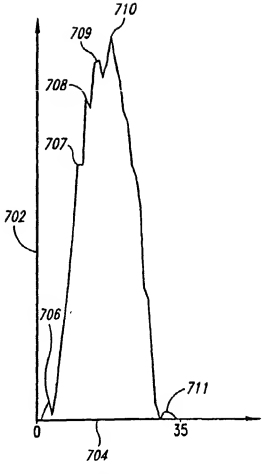


Fig. 7

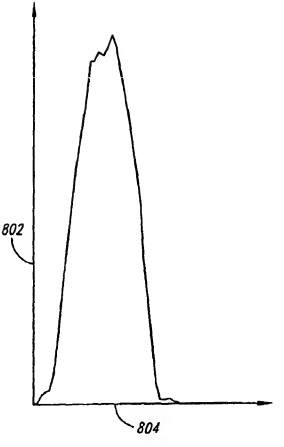


Fig. 8

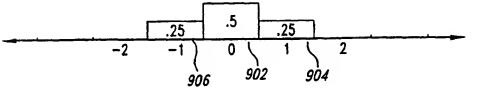


Fig. 9

Fig. 10A

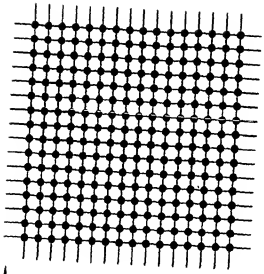


Fig. 10B

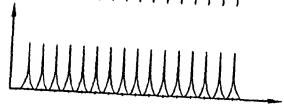


Fig. 11A

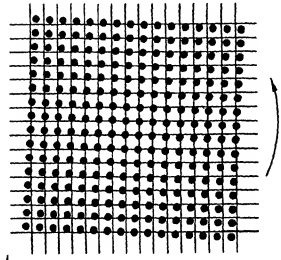
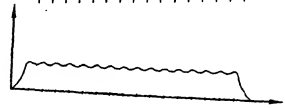


Fig. 11B



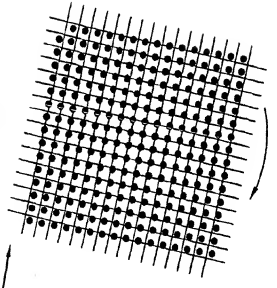


Fig. 12A

Fig. 12B

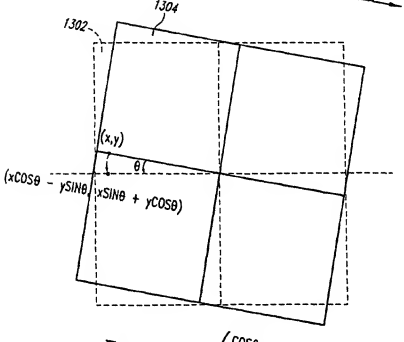


Fig. 13 $\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$

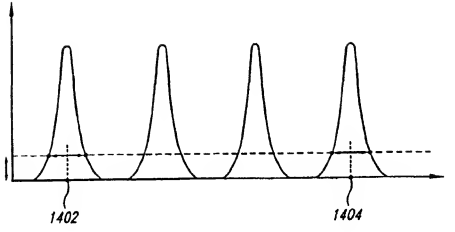


Fig. 14

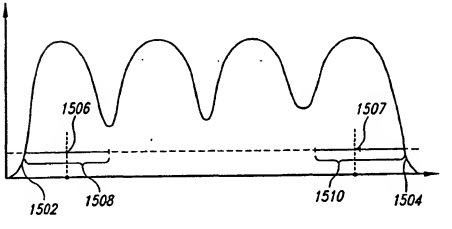


Fig. 15

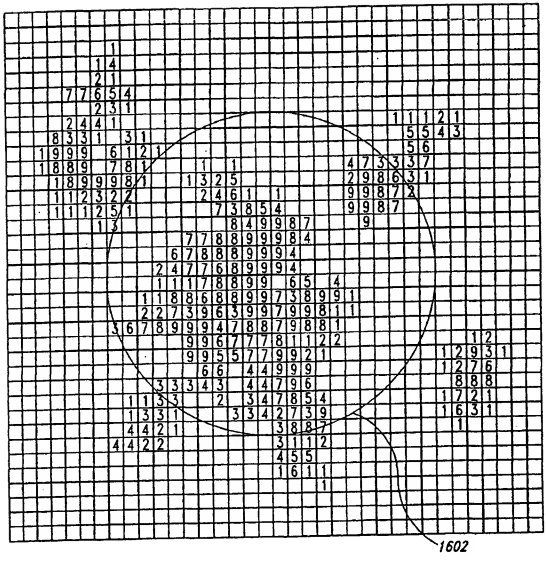


Fig. 16

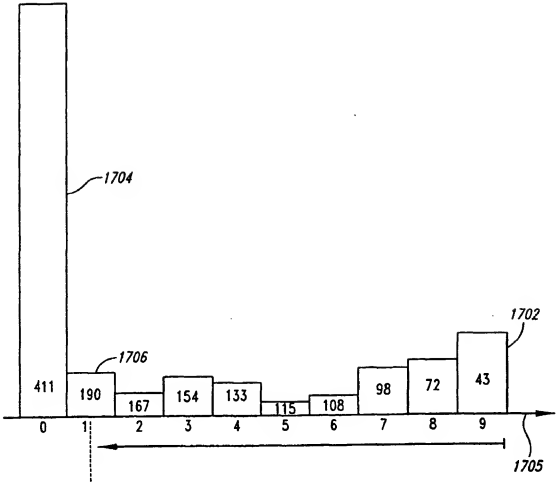

1704
1706
1702
411
190
167
154
133
115
108
98
72
43
0
1
2
3
4
5
6
7
8
9
1705

Fig. 17

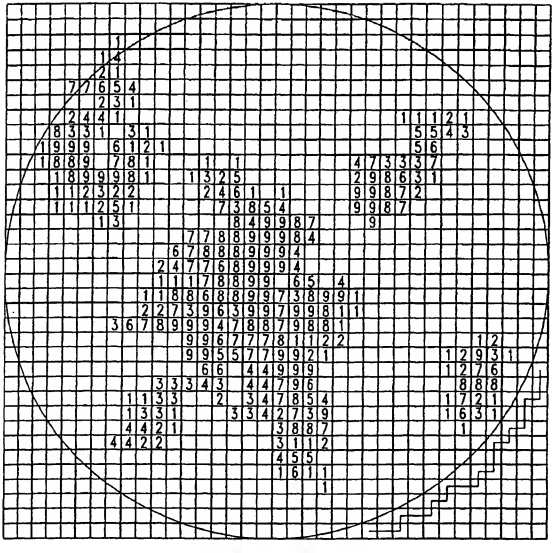


Fig. 18

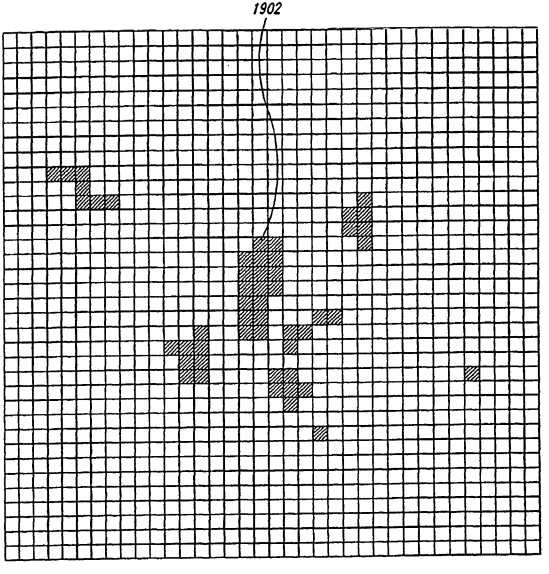


Fig. 19

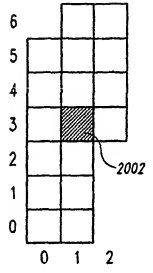

6
5
4
3
2
1
0
0 1 2
2002

Fig. 20

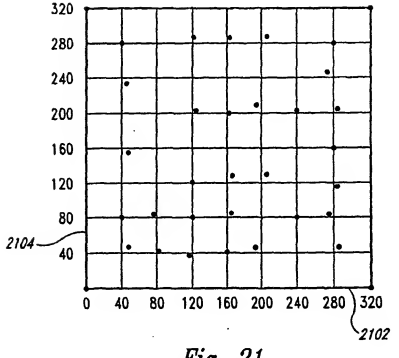

320
280
240
200
160
120
80
40
2104
0 40 80 120 160 200 240 280 320
2102

Fig. 21

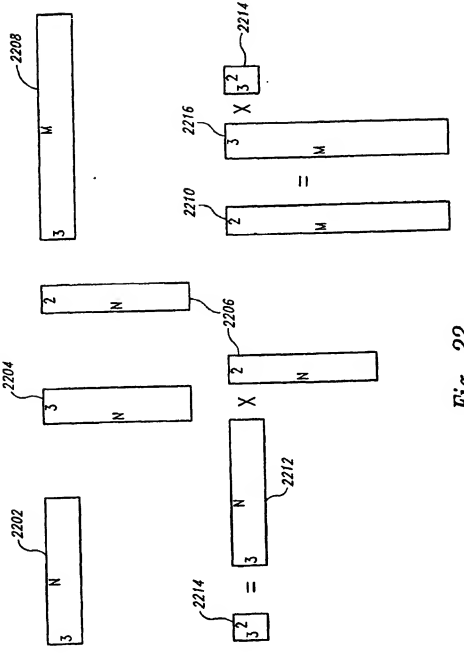

2208
3
M
2214
3^2
2216
3
X
M
=
2210
2
M
2206
2
N
2204
3
N
X
2202
N
3
2212
N
3
=
2214
3^2

Fig. 22

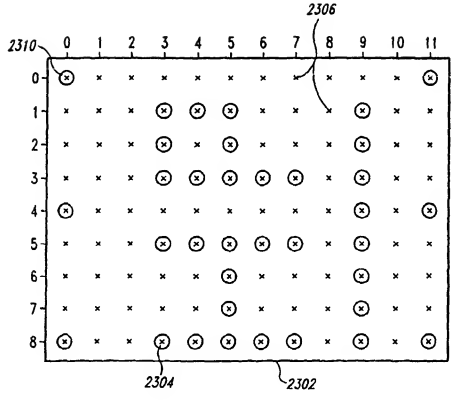


Fig. 23A

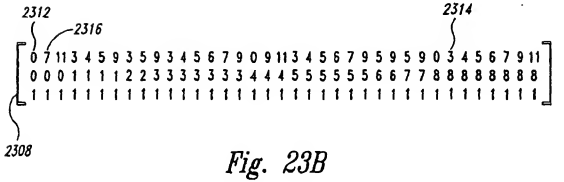


Fig. 23B

-0.017451	-0.016036	0.198716
0.004029	-0.016920	0.076562
0.016303	-0.017425	0.006760
-0.008371	-0.012535	0.130329
-0.005303	-0.012661	0.112878
-0.002234	-0.012788	0.095428
0.010040	-0.013293	0.025625
-0.008497	-0.008656	0.114293
-0.002360	-0.008908	0.079392
0.009914	-0.009413	0.009590
-0.008624	-0.004776	0.098257
-0.005555	-0.004902	0.080807
-0.002487	-0.005029	0.063356
0.000582	-0.005155	0.045905
0.003650	-0.005281	0.028455
0.009788	-0.005534	-0.006446
-0.017956	-0.000518	0.134573
0.009661	-0.001654	-0.022482
0.015798	-0.001907	-0.057383
-0.008876	0.002983	0.066185
-0.005808	0.002857	0.048735
-0.002739	0.002731	0.031284
0.000329	0.002604	0.013834
0.003398	0.002478	-0.003617
0.009535	0.002225	-0.038518
-0.002865	0.006610	0.015249
0.009409	0.006105	-0.054554
-0.002992	0.010490	-0.000787
0.009282	0.009985	-0.070589
-0.018461	0.015001	0.070430
-0.009255	0.014622	0.018078
-0.006187	0.014495	0.000628
-0.003118	0.014369	-0.016823
-0.000049	0.014243	-0.034274
0.003019	0.014117	-0.051724
0.009156	0.013864	-0.086625
0.015293	0.013611	-0.121526

Fig. 24

86.089577	26.017683
383.000000	31.000000
551.291059	26.359085
212.931324	68.929091
255.231273	69.194722
297.287336	69.211879
466.739268	68.997430
212.698809	111.308354
297.171645	111.213493
466.523521	111.238239
212.651844	153.379944
254.917398	153.737417
297.073053	153.872811
339.361872	154.102630
381.928982	154.162067
466.615409	153.735791
85.775819	195.305283
466.525669	195.796267
551.399497	195.886935
212.009756	238.402912
254.664292	238.423417
296.787962	238.298611
339.073554	238.397928
381.649538	238.367180
466.396305	238.098358
296.617866	281.180204
466.413707	280.707080
296.652245	323.094132
466.344320	322.877040
85.168403	364.952355
212.153659	365.236585
254.556017	365.400005
296.752238	365.379902
339.075490	365.433413
381.556377	365.535990
466.149127	365.083226
551.117647	365.164216

Fig. 25

42.336575	0.032847
-0.101626	42.265883
85.880412	26.859797

Fig. 26

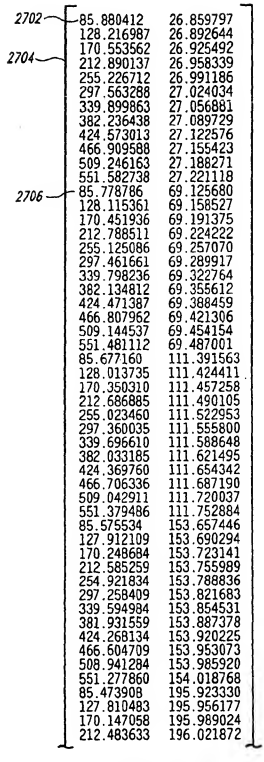

2702
2704
2706
85.880412 26.859797
128.216987 26.892644
170.553562 26.925492
212.890137 26.958339
255.226712 26.991186
297.563288 27.024034
339.899863 27.056881
382.236438 27.089729
424.573013 27.122576
466.909588 27.155423
509.246163 27.188271
551.582738 27.221118
85.778786 69.125680
128.115361 69.158527
170.451936 69.191375
212.788511 69.224222
255.125086 69.257070
297.461661 69.289917
339.798236 69.322764
382.134812 69.355612
424.471387 69.388459
466.807962 69.421306
509.144537 69.454154
551.481112 69.487001
85.677160 111.391563
128.013735 111.424411
170.350310 111.457258
212.686885 111.490105
255.023460 111.522953
297.360035 111.555800
339.696610 111.588648
382.033185 111.621495
424.369760 111.654342
466.706336 111.687190
509.042911 111.720037
551.379486 111.752884
85.575534 153.657446
127.912109 153.690294
170.248684 153.723141
212.585259 153.755989
254.921834 153.788836
297.258409 153.821683
339.594984 153.854531
381.931559 153.887378
424.268134 153.920225
466.604709 153.953073
508.941284 153.985920
551.277860 154.018768
85.473908 195.923330
127.810483 195.956177
170.147058 195.989024
212.483633 196.021872

Fig. 27A

254.820208	196.054719
297.156783	196.087566
339.493358	196.120414
381.829933	196.153261
424.166508	196.186109
466.503083	196.218956
508.839658	196.251803
551.176233	196.284651
85.372282	238.189213
127.708857	238.222060
170.045432	238.254907
212.382007	238.287755
254.718582	238.320602
297.055157	238.353450
339.391732	238.386297
381.728307	238.419144
424.064882	238.451992
466.401457	238.484839
508.738032	238.517687
551.074607	238.550534
85.270655	280.455096
127.607230	280.487943
169.943806	280.520791
212.280381	280.553638
254.616956	280.586485
296.953531	280.619333
339.290106	280.652180
381.626681	280.685028
423.963256	280.717875
466.299831	280.750722
508.636406	280.783570
550.972981	280.816417
85.169029	322.720979
127.505604	322.753826
169.842179	322.786674
212.178754	322.819521
254.515330	322.852369
296.851905	322.885216
339.188480	322.918063
381.525055	322.950911
423.861630	322.983758
466.198205	323.016606
508.534780	323.049453
550.871355	323.082300
85.067403	364.986862
127.403978	365.019710
169.740553	365.052557
212.077128	365.085404
254.413703	365.118252
296.750278	365.151099
339.086854	365.183947
381.423429	365.216794
423.760004	365.249641
466.096579	365.282489
508.433154	365.315336
550.769729	365.348184

Fig. 27B

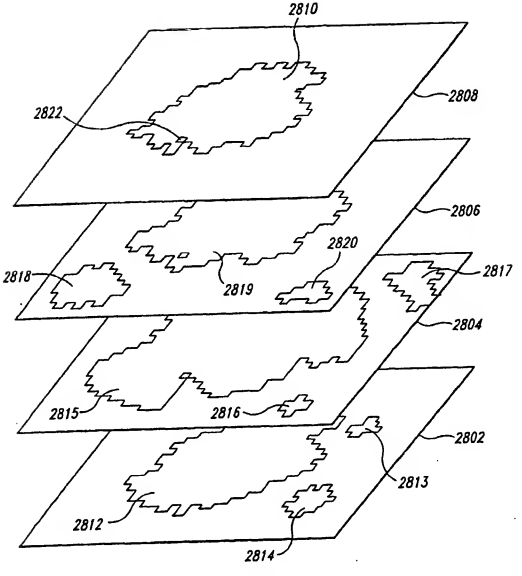

2810
2808
2822
2806
2820
2818
2819
2817
2804
2815
2816
2802
2813
2812
2814

Fig. 28

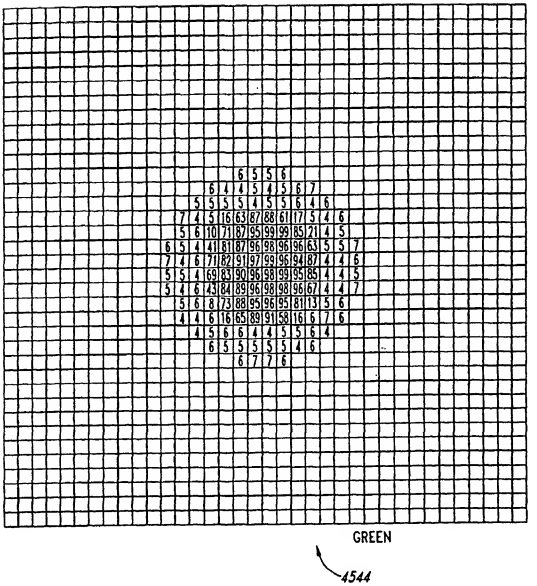


Fig. 29

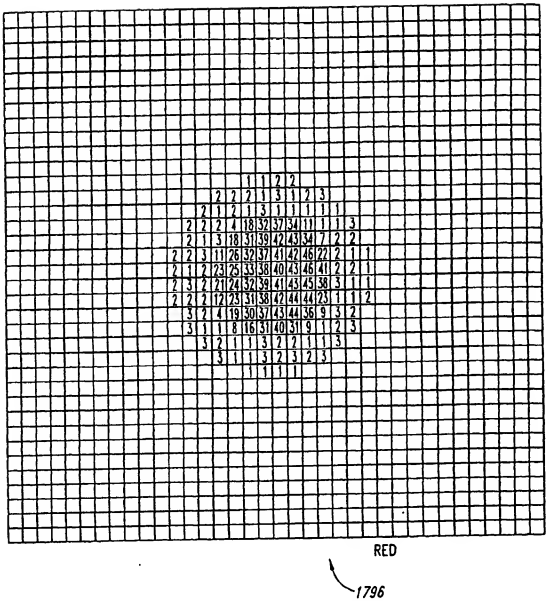


Fig. 30

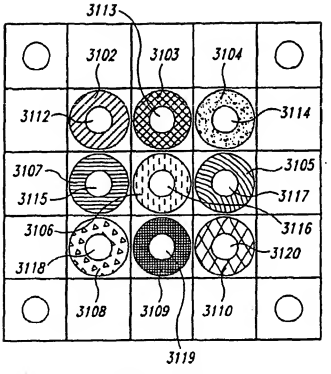

3113
3102
3103
3104
3112
3114
3107
3105
3115
3117
3106
3116
3118
3120
3108
3109
3110
3119

Fig. 31

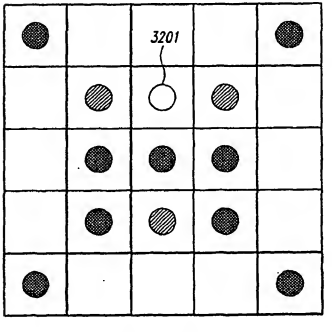

3201

Fig. 32

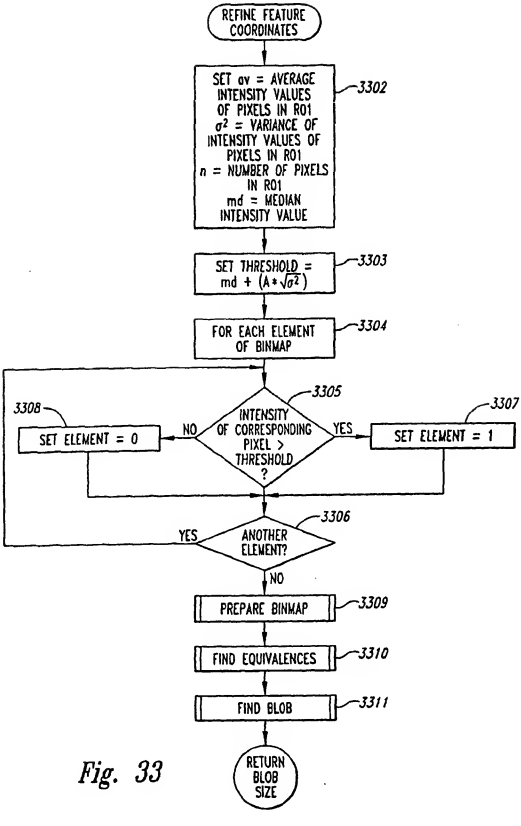

REFINE FEATURE COORDINATES
SET av = AVERAGE INTENSITY VALUES OF PIXELS IN ROI
σ² = VARIANCE OF INTENSITY VALUES OF PIXELS IN ROI
n = NUMBER OF PIXELS IN ROI
md = MEDIAN INTENSITY VALUE
3302
SET THRESHOLD = md + (A * √σ²)
3303
FOR EACH ELEMENT OF BINMAP
3304
INTENSITY OF CORRESPONDING PIXEL > THRESHOLD ?
3305
NO
YES
3308
SET ELEMENT = 0
3307
SET ELEMENT = 1
ANOTHER ELEMENT?
3306
YES
NO
PREPARE BINMAP
3309
FIND EQUIVALENCES
3310
FIND BLOB
3311
RETURN BLOB SIZE

Fig. 33

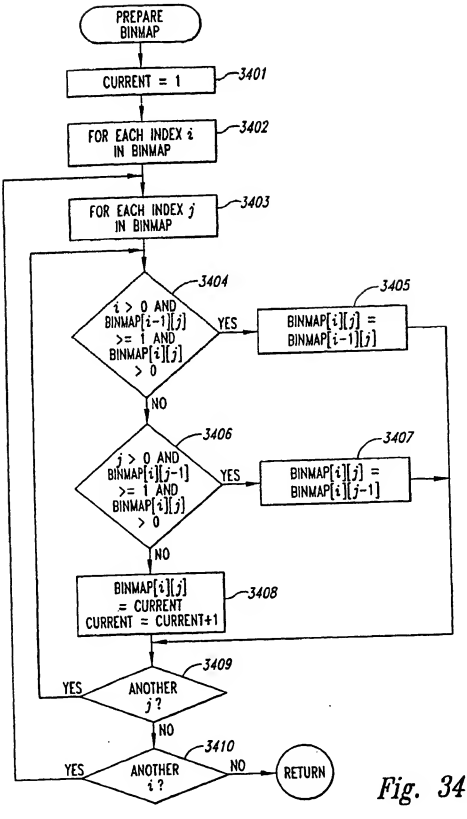

PREPARE BINMAP
CURRENT = 1
3401
FOR EACH INDEX i IN BINMAP
3402
FOR EACH INDEX j IN BINMAP
3403
i > 0 AND BINMAP[i-1][j] >= 1 AND BINMAP[i][j] > 0
3404
YES
BINMAP[i][j] = BINMAP[i-1][j]
3405
NO
j > 0 AND BINMAP[i][j-1] >= 1 AND BINMAP[i][j] > 0
3406
YES
BINMAP[i][j] = BINMAP[i][j-1]
3407
NO
BINMAP[i][j] = CURRENT CURRENT = CURRENT+1
3408
ANOTHER j ?
3409
YES
NO
ANOTHER i ?
3410
YES
NO
RETURN

Fig. 34

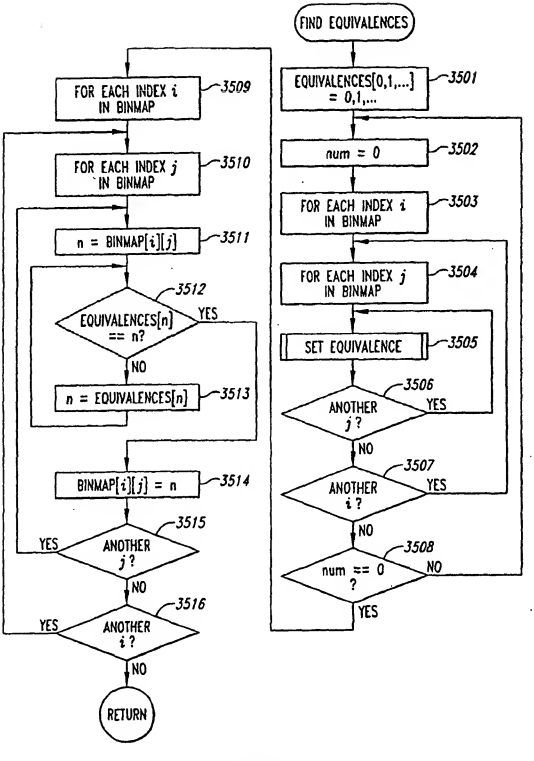

FIND EQUIVALENCES
EQUIVALENCES[0,1,...] = 0,1,...
3501
num = 0
3502
FOR EACH INDEX i IN BINMAP
3503
FOR EACH INDEX j IN BINMAP
3504
SET EQUIVALENCE
3505
ANOTHER j ?
3506
YES
NO
ANOTHER i ?
3507
YES
NO
num == 0 ?
3508
NO
YES
FOR EACH INDEX i IN BINMAP
3509
FOR EACH INDEX j IN BINMAP
3510
n = BINMAP[i][j]
3511
EQUIVALENCES[n] == n?
3512
YES
NO
n = EQUIVALENCES[n]
3513
BINMAP[i][j] = n
3514
ANOTHER j ?
3515
YES
NO
ANOTHER i ?
3516
YES
NO
RETURN

Fig. 35

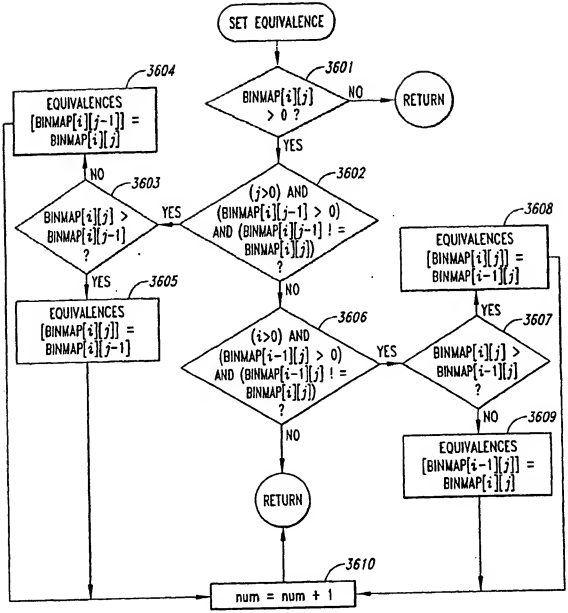

SET EQUIVALENCE
3601
BINMAP[i][j] > 0 ?
NO
RETURN
YES
3602
(j>0) AND (BINMAP[i][j-1] > 0) AND (BINMAP[i][j-1] ! = BINMAP[i][j]) ?
3603
BINMAP[i][j] > BINMAP[i][j-1] ?
YES
NO
3604
EQUIVALENCES [BINMAP[i][j-1]] = BINMAP[i][j]
3605
EQUIVALENCES [BINMAP[i][j]] = BINMAP[i][j-1]
NO
3606
(i>0) AND (BINMAP[i-1][j] > 0) AND (BINMAP[i-1][j] ! = BINMAP[i][j]) ?
YES
3607
BINMAP[i][j] > BINMAP[i-1][j] ?
YES
NO
3608
EQUIVALENCES [BINMAP[i][j]] = BINMAP[i-1][j]
3609
EQUIVALENCES [BINMAP[i-1][j]] = BINMAP[i][j]
NO
RETURN
3610
num = num + 1

Fig. 36

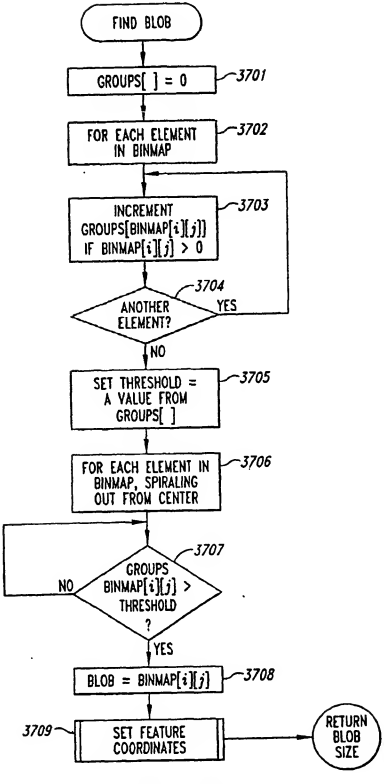

FIND BLOB
GROUPS[ ] = 0
3701
FOR EACH ELEMENT IN BINMAP
3702
INCREMENT GROUPS[BINMAP[i][j]] IF BINMAP[i][j] > 0
3703
ANOTHER ELEMENT?
3704
YES
NO
SET THRESHOLD = A VALUE FROM GROUPS[ ]
3705
FOR EACH ELEMENT IN BINMAP, SPIRALING OUT FROM CENTER
3706
GROUPS BINMAP[i][j] > THRESHOLD ?
3707
NO
YES
BLOB = BINMAP[i][j]
3708
3709
SET FEATURE COORDINATES
RETURN BLOB SIZE

Fig. 37

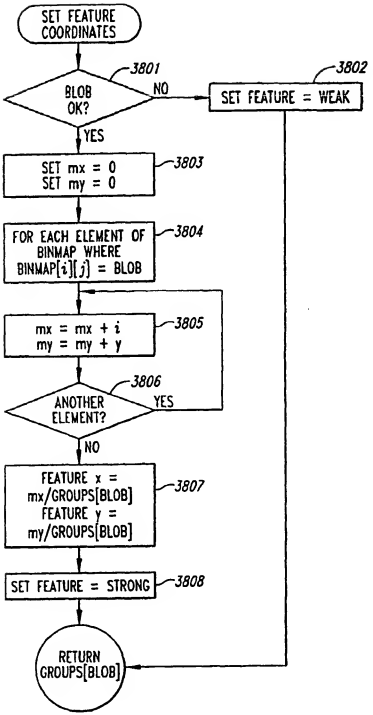

SET FEATURE COORDINATES
3801
BLOB OK?
NO
3802
SET FEATURE = WEAK
YES
SET mx = 0
SET my = 0
3803
FOR EACH ELEMENT OF BINMAP WHERE BINMAP[i][j] = BLOB
3804
mx = mx + i
my = my + y
3805
3806
ANOTHER ELEMENT?
YES
NO
FEATURE x = mx/GROUPS[BLOB]
FEATURE y = my/GROUPS[BLOB]
3807
SET FEATURE = STRONG
3808
RETURN GROUPS[BLOB]

Fig. 38

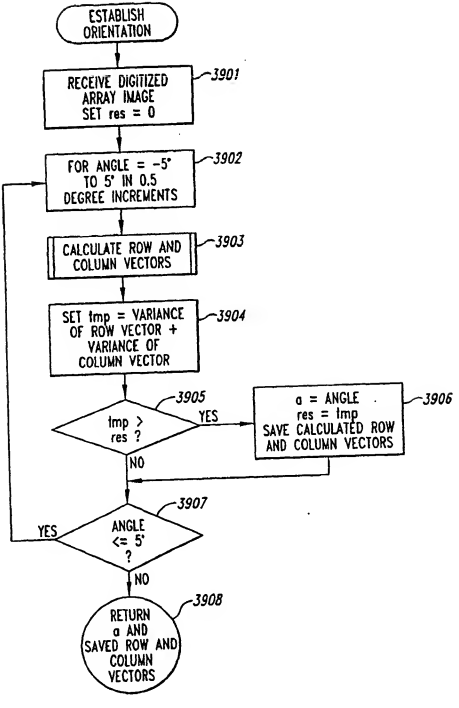

ESTABLISH ORIENTATION
RECEIVE DIGITIZED ARRAY IMAGE SET res = 0
3901
FOR ANGLE = -5° TO 5° IN 0.5 DEGREE INCREMENTS
3902
CALCULATE ROW AND COLUMN VECTORS
3903
SET tmp = VARIANCE OF ROW VECTOR + VARIANCE OF COLUMN VECTOR
3904
tmp > res ?
3905
YES
NO
α = ANGLE res = tmp SAVE CALCULATED ROW AND COLUMN VECTORS
3906
ANGLE <= 5° ?
3907
YES
NO
RETURN α AND SAVED ROW AND COLUMN VECTORS
3908

Fig. 39

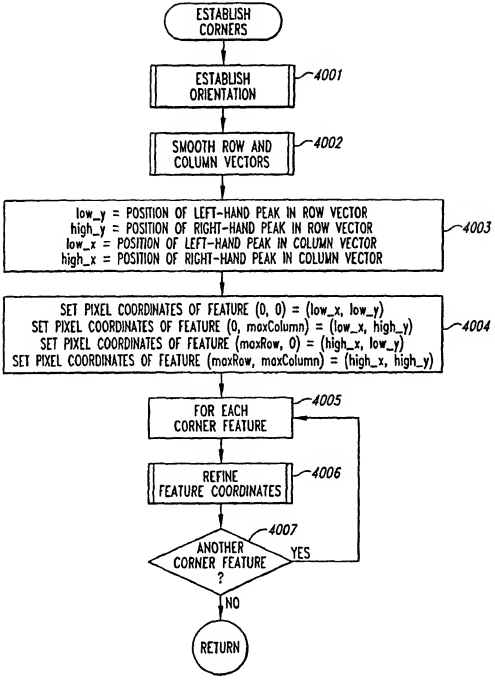
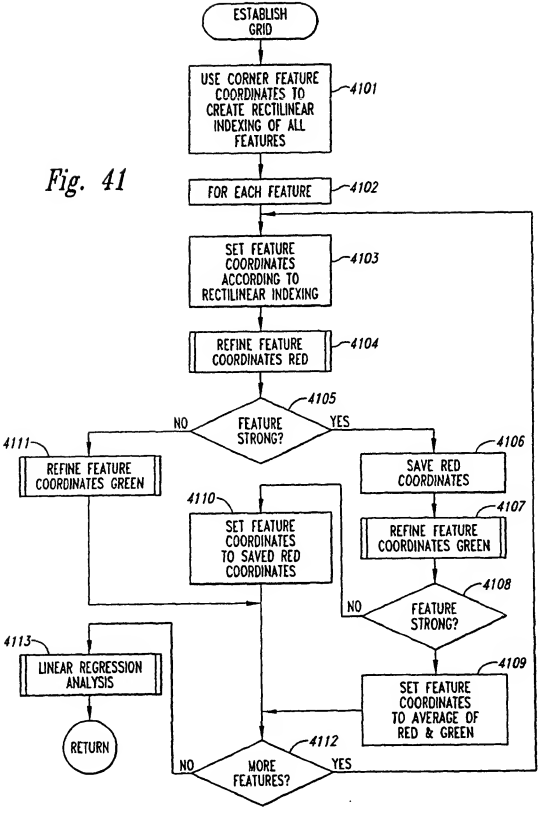

ESTABLISH CORNERS
ESTABLISH ORIENTATION
4001
SMOOTH ROW AND COLUMN VECTORS
4002
low_y = POSITION OF LEFT-HAND PEAK IN ROW VECTOR
high_y = POSITION OF RIGHT-HAND PEAK IN ROW VECTOR
low_x = POSITION OF LEFT-HAND PEAK IN COLUMN VECTOR
high_x = POSITION OF RIGHT-HAND PEAK IN COLUMN VECTOR
4003
SET PIXEL COORDINATES OF FEATURE (0, 0) = (low_x, low_y)
SET PIXEL COORDINATES OF FEATURE (0, maxColumn) = (low_x, high_y)
SET PIXEL COORDINATES OF FEATURE (maxRow, 0) = (high_x, low_y)
SET PIXEL COORDINATES OF FEATURE (maxRow, maxColumn) = (high_x, high_y)
4004
FOR EACH CORNER FEATURE
4005
REFINE FEATURE COORDINATES
4006
ANOTHER CORNER FEATURE ?
4007
YES
NO
RETURN

Fig. 40

Fig. 41


ESTABLISH GRID
USE CORNER FEATURE COORDINATES TO CREATE RECTILINEAR INDEXING OF ALL FEATURES
4101
FOR EACH FEATURE
4102
SET FEATURE COORDINATES ACCORDING TO RECTILINEAR INDEXING
4103
REFINE FEATURE COORDINATES RED
4104
4105
FEATURE STRONG?
NO
YES
4111
REFINE FEATURE COORDINATES GREEN
4106
SAVE RED COORDINATES
4110
SET FEATURE COORDINATES TO SAVED RED COORDINATES
4107
REFINE FEATURE COORDINATES GREEN
4108
NO
FEATURE STRONG?
4113
LINEAR REGRESSION ANALYSIS
4109
SET FEATURE COORDINATES TO AVERAGE OF RED & GREEN
RETURN
4112
NO
MORE FEATURES?
YES